

18.041/H/03

TUGAS AKHIR  
KS 1701



**STUDI PENERAPAN KENDALI LOGIKA FUZZY  
PADA SISTEM OLAH GERAK ROV**



RSSP  
SII.3  
Djo  
S-1  
2003

OLEH :

**DJOKO WAHYUDI**  
NRP. 4298 100 001

**JURUSAN TEKNIK SISTEM PERKAPALAN  
FAKULTAS TEKNOLOGI KELAUTAN  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2003**

PERPUSTAKAAN ITS	
Tgl. Terima	10-4-2003
Terima Dari	H



**TUGAS AKHIR**  
**KS 1701**

**STUDI PENERAPAN KENDALI LOGIKA FUZZY  
PADA SISTEM OLAH GERAK ROV**



**OLEH :**

**DJOKO WAHYUDI**  
**NRP. 4298 100 001**

**JURUSAN TEKNIK SISTEM PERKAPALAN  
FAKULTAS TEKNOLOGI KELAUTAN  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2003**



**STUDI PENERAPAN KENDALI LOGIKA FUZZY  
PADA SISTEM OLAH GERAK ROV**

**TUGAS AKHIR**

**DIAJUKAN GUNA MEMENUHI SEBAGIAN PERSYARATAN  
UNTUK MEMPEROLEH GELAR SARJANA  
PADA  
TEKNIK SISTEM PERKAPALAN  
FAKULTAS TEKNOLOGI KELAUTAN  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA**

**Mengetahui/Menyetujui,**

**DOSEN PEMBIMBING I**

Ir. I MADE ARIANA, MT.  
NIP. 132 133 971

**DOSEN PEMBIMBING II**

Ir. SARDONO SARWITO, MSc.  
NIP. 131 651 255

**SURABAYA  
JANUARI, 2003**





**FAKULTAS TEKNOLOGI KELAUTAN**  
**JURUSAN TEKNIK SISTEM PERKAPALAN**

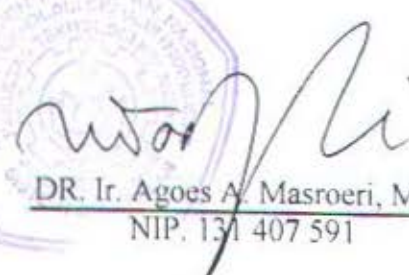
KAMPUS ITS KEPUTIH SUKOLILI SURABAYA 60111  
TELP. 5994754, 5994251 – 55 PES. 1102 FAX 5994754

**SURAT KEPUTUSAN Pengerjaan Tugas Akhir KS 1701**


Sebagai salah satu syarat untuk mendapatkan gelar Sarjana Teknik Sistem Perkapalan Fakultas Teknologi Kelautan ITS, maka perlu ditertibkan Surat Keputusan Pengerjaan Tugas Akhir yang memberikan tugas kepada mahasiswa tersebut di bawah untuk mengerjakan Tugas sesuai judul dan lingkup bahasan yang ditentukan.


Nama Mahasiswa : DJOKO WAHYUDI  
NRP. : 4298 100 001  
Dosen Pembimbing : 1. Ir. I MADE ARIANA, MT.  
2. Ir. SARDONO SARWITO, MSc.  
Tanggal Diberikan Tugas : Oktober 2002  
Tanggal Diselesaikan Tugas :  
Judul Tugas Akhir : STUDI PENERAPAN KENDALI LOGIKA  
FUZZY PADA SISTEM OLAH GERAK ROV


Surabaya, Oktober 2002  
Ketua Jurusan Teknik Sistem Perkapalan  
FT. Kelautan ITS

  
DR. Ir. Agoes A. Masroeri, M.Eng  
NIP. 131 407 591

Surabaya, Oktober 2002  
Yang menerima tugas:

Mahasiswa  
  
Djoko Wahyudi  
NRP. 4298 100 001

Dosen Pembimbing I  
  
Ir. I Made Ariana, MT.  
NIP. 132 133 971

Dosen Pembimbing II  
  
Ir. Sardono Sarwito, MSc.  
NIP. 131 651 255





## ABSTRAK

Dalam sistem pengendalian pada sistem propulsi Remotely Operated Vehicle (ROV) dapat dilakukan dengan metode-metode kontrol. Namun dalam penulisan tugas akhir ini akan dicoba pengendalian dengan menggunakan Kendali Logika Fuzzy (KLF). Dasar pemikiran ini diambil dengan bertitik tolak pada beberapa pertimbangan yaitu dilihat dari keunggulan KLF. Dalam konsep KLF tidak dituntut pemodelan matematik yang dalam kenyataannya cukup rumit, namun informasi cukup didapat dari hasil pengalaman operator.

Dengan menerapkan KLF pada sistem olah gerak ROV diharapkan gerakan ROV tetap stabil dalam mengatasi perubahan kondisi/bahaya lingkungan dimana ROV tersebut beroperasi.

Semakin besar variabel masukan yang diterima, baik untuk variabel masukan sudut posisi, sudut rolling, dan sudut pitching, semakin besar kebutuhan perbedaan thrust yang diperlukan tetapi lama waktu yang diperlukan untuk mencapai kondisi yang diinginkan tetap sama untuk tiap-tiap variabel masukan.



## KATA PENGANTAR

Segala puji dan syukur Alhamdulillah ke hadirat ALLAH SWT karena anugerah dan bimbingan-Nya, maka seluruh rangkaian kegiatan perkuliahan selama berada di lingkungan Institut ataupun dalam mengaplikasikannya di lapangan sampai penulisan Tugas Akhir ini dapat penulis selesaikan dengan baik.

Hasil dari penulisan ini diharapkan dapat dijadikan sebagai salah satu referensi bagi mahasiswa pada umumnya dan mahasiswa Teknik Sistem Perkapalan pada khususnya untuk menambah wawasan pengetahuan tentang sistem kontrol logika fuzzy pada olah gerak rov ataupun pihak-pihak lain yang terkait langsung maupun tidak langsung supaya dapat memahami fungsi dan peranan sistem kontrol logika fuzzy dalam dunia permesinan perkapalan.

Penulis menyadari bahwa penulisan ini masih jauh dari kesempurnaan, oleh karena itu dengan kerendahan hati penulis menerima saran dan petunjuk untuk kebaikan penulisan ini.

Akhirnya semoga penulisan ini dapat bermanfaat bagi yang memerlukannya.

Surabaya, Januari 2003

Penulis,



## UCAPAN TERIMA KASIH

Penyusunan Tugas Akhir ini dapat berlangsung atas bantuan dan uluran tangan dari berbagai pihak. Untuk itu pada kesempatan ini penulis menyampaikan terima kasih yang tak terhingga kepada :

1. Dr. Ir. A.A. Masroeri, M. Eng, selaku Ketua Jurusan Teknik Sistem Perkapalan.
2. Ir. I Made Ariana, MT, selaku Dosen Pembimbing I yang banyak memberikan arahan-arahan dalam penyelesaian Tugas Akhir.
3. Ir. Sardono Sarwito, MSc, selaku Dosen Pembimbing II yang banyak memberikan arahan-arahan dalam penyelesaian Tugas Akhir.
4. Ir. Bambang Supangkat, selaku Dosen Wali penulis.
5. Bapak (Alm. Sahad), Ibu (Kasijami), adik (Yudha Sangputro), Koko Sudarno, serta seluruh keluarga dan saudara yang banyak memberikan bantuan baik secara spiritual maupun material yang tak ternilai.
6. Mas Amirul, atas bantuannya membuatkan program.
7. Sesama pengambil tugas akhir tentang fuzzy logic, Arief Kusbyanto '98, Sofyan Hadi '98, atas diskusi dan dukungannya.
8. Semua angkatan '98 Teknik Sistem Perkapalan, FTK – ITS terutama Heru SB '98, Nono '98, Dudi '96 atas bantuannya selama ini serta seluruh civitas lab. Reliability atas kebersamaannya mengerjakan tugas akhir ini sampai pagi.
9. Semua pihak baik secara langsung maupun tidak langsung membantu penulis menyelesaikan tugas akhir ini.





Semoga ALLAH SWT membalas segala budi baik yang telah diberikan kepada penulis. Amien.

Surabaya, Januari 2003

Penulis,



## DAFTAR ISI

HALAMAN JUDUL .....	i
LEMBAR PENGESAHAN .....	ii
SURAT KEPUTUSAN .....	iii
ABSTRAK .....	iv
KATA PENGANTAR .....	v
UCAPAN TERIMA KASIH .....	vi
DAFTAR ISI .....	viii
DAFTAR TABEL .....	xii
DAFTAR GAMBAR .....	xiii
DAFTAR GRAFIK .....	xv
DAFTAR NOTASI .....	xvi
BAB I     PENDAHULUAN .....	I-1
1.1    Latar Belakang .....	I-1
1.2    Permasalahan .....	I-2
1.3    Metodologi .....	I-4
1.4    Tujuan dan Manfaat .....	I-4
1.5    Sistematika Penulisan .....	I-5
BAB II    DASAR TEORI .....	II-1
2.1    Logika Fuzzy .....	II-1
2.1.1 Pandangan Mengenai Logika Fuzzy .....	II-4
2.1.2 Kelebihan Menggunakan Logika Fuzzy .....	II-7



2.1.3	Logika Fuzzy Tidak Digunakan .....	II-8
2.2	Teori Himpunan Fuzzy .....	II-9
2.2.1	Himpunan Fuzzy .....	II-9
2.2.2	Fungsi Keanggotaan Himpunan Fuzzy .....	II-10
2.2.3	Variabel Linguistik .....	II-12
2.2.4	Operasi Dasar Himpunan Fuzzy .....	II-14
2.3	Struktur Dasar Unit Kendali Logika Fuzzy .....	II-15
2.3.1	Fuzzifikasi .....	II-16
2.3.2	Inferensi Fuzzy (Fuzzy Inference Rules) .....	II-17
2.3.3	Basis Pengetahuan Fuzzy .....	II-19
2.3.4	Penurunan Aturan Kontrol Fuzzy .....	II-20
2.3.5	Teknik Penalaran Fuzzy (Fuzzy Reasoning) .....	II-24
2.3.6	Defuzzifikasi .....	II-25
2.4	Kontrol Subsea (Subsea Controls) .....	II-26
2.4.1	Tipe Kontrol Subsea .....	II-28
2.5	ROV .....	II-31
2.5.1	Sejarah Singkat ROV .....	II-33
2.5.2	Tipe ROV .....	II-34
2.5.3	Daerah Kerja ROV .....	II-35
2.6	Design ROV .....	II-37
2.6.1	Sistem Kontrol .....	II-39
2.6.2	Propulsi .....	II-40
2.7	Pengoperasian Umum ROV .....	II-40





BAB III	PEMODELAN SISTEM PROPULSI ROV .....	III-1
3.1	Tinjauan Umum Sistem .....	III-1
3.2	Arsitektur Sistem Kontrol .....	III-6
BAB IV	PERANCANGAN ALGORITMA KENDALI LOGIKA FUZZY .....	IV-1
4.1	Arsitektur Pengembangan Sistem Pengendalian .....	IV-1
4.2	Perancangan Kendali Logika Fuzzy .....	IV-2
4.2.1	Penentuan Masukan dan Keluaran Sistem .....	IV-2
4.2.2	Struktur Sistem Logika Fuzzy .....	IV-2
4.2.3	Penentuan Sifat-Sifat Variabel Linguistik .....	IV-3
4.2.4	Fungsi Keanggotaan .....	IV-4
4.2.4.1	Fungsi Keanggotaan Variabel Sudut Posisi .....	IV-4
4.2.4.2	Fungsi Keanggotaan Variabel Sudut Rolling .....	IV-5
4.2.4.3	Fungsi Keanggotaan Variabel Sudut Pitching .....	IV-6
4.2.4.4	Fungsi Keanggotaan Variabel Delta1 .....	IV-8
4.2.4.5	Fungsi Keanggotaan Variabel Delta2 .....	IV-9
4.2.4.6	Fungsi Keanggotaan Variabel Delta3 .....	IV-10
4.2.5	Pengembangan Rule Base .....	IV-10
4.3	Sistem Penentuan Thrust dari Masukan .....	IV-17
BAB V	ANALISA HASIL PERANCANGAN .....	V-1
5.1	Simulasi .....	V-1
5.1.1	Simulasi Terhadap Sudut Posisi (SP) .....	V-5
5.1.2	Simulasi Terhadap Sudut Rolling (SR) .....	V-8
5.1.3	Simulasi Terhadap Sudut Pitching (SPt) .....	V-10



5.2	Analisa .....	V-13
5.2.1	Analisa Terhadap Sudut Posisi .....	V-14
5.2.2	Analisa Terhadap Sudut Rolling .....	V-16
5.2.3	Analisa Terhadap Sudut Pitching .....	V-19
BAB VI	PENUTUP .....	VI-1
6.1	Kesimpulan .....	VI-1
6.2	Saran .....	VI-1
DAFTAR PUSTAKA .....		xix
LAMPIRAN .....		xxi



## DAFTAR TABEL

No. Tab.	Keterangan	Hal.
2.1	Prototipe aturan kontrol linguistik dengan 3 nilai	II-21
2.2	Penyempurnaan aturan kontrol dengan 3 nilai	II-22
2.3	Prototipe aturan kontrol dengan 7 nilai	II-23
2.4	Penyempurnaan aturan kontrol dengan 7 nilai	II-23
2.5	Keputusan aturan kontrol fuzzy Mac Vicar – Whelan	II-24
4.1	Sifat variabel linguistik	IV-3
4.2	Detail variabel linguistik	IV-4
4.3	Titik batas keanggotaan sudut posisi	IV-5
4.4	Titik batas keanggotaan sudut rolling	IV-6
4.5	Titik batas keanggotaan sudut pitching	IV-7
4.6	Titik batas keanggotaan delta1	IV-8
4.7	Titik batas keanggotaan delta2	IV-9
4.8	Titik batas keanggotaan delta3	IV-10
4.9	Rule base	IV-15
5.1	Tabel simulasi terhadap sudut posisi	V-8
5.2	Tabel simulasi terhadap sudut rolling	V-11
5.3	Tabel simulasi terhadap sudut pitching	V-15
5.4	Tabel simulasi terhadap variasi sudut posisi	V-19
5.5	Tabel simulasi terhadap variasi sudut rolling	V-23
5.6	Tabel simulasi terhadap variasi sudut pitching	V-27





## DAFTAR GAMBAR

No. Gb.	Keterangan	Hal.
2.1	Perbandingan metode konvensional dan fuzzy	II-2
2.2	Perbedaan ketelitian dan ketepatan dalam dunia nyata	II-5
2.3	Pemetaan daerah masukan ke daerah keluaran	II-6
2.4	Bentuk-bentuk fungsi keanggotaan	II-12
2.5	Penafsiran grafik variabel linguistik	II-13
2.6	Struktur dasar kendali logika fuzzy	II-16
2.7	Tanggapan sistem loop tertutup	II-21
2.8	Aturan Inferensi Minimum – Mamdani	II-25
2.9	Alternatif pemilihan sistem kontrol	II-27
2.10	Sistem kontrol electro-hydraulic	II-30
2.11	Launch system ROV	II-32
2.12	Remote operated vehicles	II-35
2.13	Pengeboran di teluk Meksiko	II-36
2.14	Design spiral ROV	II-38
3.1	Performance thruster tipe HT 380 BA-32	III-3
3.2	Performance thruster tipe HT 300 BA-32	III-5
3.3	Penempatan thruster ROV	III-6
3.4	Arsitektur sistem kontrol ROV	III-7
4.1	Arsitektur sistem pengendalian ROV	IV-1
4.2	Struktur sistem logika fuzzy	IV-3



4.3	Fungsi keanggotaan sudut posisi	IV-4
4.4	Fungsi keanggotaan sudut rolling	IV-6
4.5	Fungsi keanggotaan sudut pitching	IV-7
4.6	Fungsi keanggotaan $\Delta 1$	IV-8
4.7	Fungsi keanggotaan $\Delta 2$	IV-9
4.8	Fungsi keanggotaan $\Delta 3$	IV-10
4.9	Tampilan rule base pada Matlab	IV-17
4.10	Metode penentuan thrust	IV-19
4.11	Tampilan metode penentuan thrust pada Matlab	IV-20
5.1	Tampilan program Matlab – Fungsi keanggotaan sudut posisi	V-2
5.2	Tampilan program Matlab – Fungsi keanggotaan sudut rolling	V-2
5.3	Tampilan program Matlab – Fungsi keanggotaan sudut pitching	V-3
5.4	Tampilan program Matlab – Fungsi keanggotaan $\Delta 1$	V-4
5.5	Tampilan program Matlab – Fungsi keanggotaan $\Delta 2$	V-4
5.6	Tampilan program Matlab – Fungsi keanggotaan $\Delta 3$	V-5
5.7	Grafik & tabel hasil simulasi sudut posisi-program delphi	V-21
5.8	Grafik & tabel hasil simulasi sudut rolling - program delphi	V-26
5.9	Grafik & tabel hasil simulasi sudut pitching - program delphi	V-31



## DAFTAR GRAFIK

Nomor Grafik	Keterangan	Halaman
5.1	Grafik simulasi terhadap sudut posisi	V-9
5.2	Grafik simulasi terhadap sudut rolling	V-12
5.3	Grafik simulasi terhadap sudut pitching	V-16
5.4	Grafik simulasi terhadap variasi sudut posisi	V-19
5.5	Grafik simulasi terhadap variasi sudut rolling	V-23
5.6	Grafik simulasi terhadap variasi sudut pitching	V-27





## DAFTAR NOTASI

- SP : sudut posisi (deg)
  - (+) : arah belok ke kanan
  - (-) : arah belok ke kiri
- SR : sudut rolling (deg)
  - (-) : berlawanan arah jarum jam
  - (-) : searah jarum jam
- SPt : sudut pitching (deg)
  - (+) : menukik ke depan
  - (-) : menukik ke belakang
- $\Delta 1$  : perbedaan thrust antara vectored horizontal thruster port dan stbd (kgf)
  - (+) : port lebih besar daripada stbd
  - (-) : stbd lebih besar daripada port
- $\Delta 2$  : perbedaan thrust antara vertical thruster port dan stbd (kgf)
  - (+) : stbd lebih besar daripada port
  - (-) : port lebih besar daripada stbd
- $\Delta 3$  : perbedaan thrust antara vertical thruster depan dan belakang (kgf)
  - (+) : depan lebih besar daripada belakang
  - (-) : belakang lebih besar daripada depan
- $r_{SP}$  = lengan momen sudut posisi (= 0.725 m)
- $r_{SR}$  = lengan momen sudut rolling (= 0.765 m)
- $r_{SPt}$  = lengan momen sudut pitching (= 1.325 m)



- $k_{zz}$  : radius girasi sudut posisi (m)
- $k_{xx}$  : radius girasi sudut rolling (m)
- $k_{yy}$  : radius girasi sudut pitching (m)
- $M_{SP}$  : momen sudut posisi (kgf.m)  
 $= \Delta 1 \times r_{SP}$
- $M_{SR}$  : momen sudut rolling (kgf.m)  
 $= \Delta 2 \times r_{SR}$
- $M_{SPt}$  : momen sudut pitching (kgf.m)  
 $= \Delta 3 \times r_{SPt}$
- $\dot{SP}$  : percepatan sudut posisi
- $\dot{SR}$  : percepatan sudut rolling
- $\dot{SP}_t$  : percepatan sudut pitching
- $\dot{SP}$  : kecepatan sudut posisi
- $\dot{SR}$  : kecepatan sudut rolling
- $\dot{SP}_t$  : kecepatan sudut pitching
- $I_z$  : momen inersia terhadap sumbu z (kg.dt<sup>2</sup>.m)
- $I_x$  : momen inersia terhadap sumbu x (kg.dt<sup>2</sup>.m)
- $I_y$  : momen inersia terhadap sumbu y (kg.dt<sup>2</sup>.m)
- $t$  : waktu (dt)
- $g$  : percepatan gravitasi (m/dt<sup>2</sup>)
- $\omega_0$  : frekuensi natural (rad/dt)
- $L$  : panjang rov (m)



- $B$  : lebar rov (m)
- $H$  : tinggi rov (m)
- $D$  : Berat rov (kg)
- $\alpha$  : water line area coefisien ( $\alpha = 1$ )
- $\delta$  : blok coefisien ( $\delta = 1$ )
- $(r - a)$  : transverse metacentric height (m)
- $(R - a)$  : longitudinal metacentric height (m)





## BAB I

### PENDAHULAN

#### 1.1 Latar Belakang

Dalam beberapa tahun terakhir ini, perkembangan ilmu pengetahuan dan teknologi semakin maju pesat, terutama bidang kontrol untuk mengendalikan suatu proses produksi pada kegiatan industri. Oleh karena proses produksi dalam industri merupakan bagian yang penting maka diperlukan sistem pengendalian yang berfungsi untuk menjaga kestabilan sistem terhadap pengaruh gangguan yang terjadi selama proses berlangsung.

Perancangan sistem pengendalian, baik yang konvensional maupun modern yang digunakan pada industri, misal kita ambil sebuah contoh pada sistem olah gerak Remotely Operated Vehicle (ROV), memerlukan pendekatan dan pemodelan matematik terhadap plantnya (*Huda, 1997*). Aturan-aturan kontrolnya diturunkan dari model matematik yang menyatakan hubungan antara masukan dan keluaran plant. Untuk proses yang tidak terlalu kompleks/masih bisanya diketahui besaran-besaran kuantitatifnya, sistem pengendalian tersebut masih dapat diterapkan dengan baik, karena persamaan matematik plant masih dapat dicari. Namun seringkali besaran-besaran kuantitatif tersebut tidak diketahui melainkan hanya berupa ungkapan-ungkapan kualitatif ataupun untuk proses yang kompleks, akan sulit untuk menentukan persamaan matematiknya. Kesulitan juga akan bertambah jika perancang memperhitungkan faktor-faktor non-linearitas dan parameter-parameter proses secara lengkap. Hal ini akan memberikan kesulitan tersendiri bagi seorang operator yang tidak memahami pemodelan matematik dari



suatu plant dan kondisi ini memungkinkan terjadinya kegagalan suatu proses diakibatkan ketidakpahaman seorang operator dalam menjalankan plant tersebut. Selain itu juga perancangan sistem pengendalian tersebut membutuhkan waktu yang lama untuk pemahaman parameter pengendalinya (*Santoso, 1996*).

Dengan alasan seperti ini mendorong digunakannya sistem pengendalian yang lebih baik tanpa perlunya pemodelan/pendekatan matematik terhadap plant, sehingga seorang operator dapat mengatur suatu proses yang sedang berlangsung tanpa perlu memahami pemodelan matematik dari suatu plant melainkan cukup menterjemahkannya berupa ungkapan-ungkapan linguistik yang menggunakan bahasa natural, seperti suhu agak panas, katup terbuka penuh, dan sebagainya.

Sistem pengendalian dengan Kendali Logika Fuzzy (KLF) yang berbasiskan pengetahuan, pengalaman, cara berpikir manusia, serta ungkapan-ungkapan linguistik dapat merepresentasikan cara kerja operator manusia tersebut (*Tomsovic, 2000*), sehingga tidak memerlukan pendekatan/pemodelan matematik dari suatu plant. Dalam penulisan tugas akhir ini akan dibahas penerapan KLF pada sistem olah gerak ROV. Dengan penerapan alternatif sistem pengendalian KLF ini diharapkan memberikan hasil yang lebih baik, seperti kemudahan KLF untuk dimengerti, fleksibel, pencapaian kondisi kestabilan dalam waktu yang lebih singkat dibanding sitem pengendalian yang digunakan saat ini, dan terutama diharapkan dapat memberikan keefektifan gerak pada ROV dalam mengatasi perubahan kondisi/bahaya lingkungan dimana ROV tersebut bekerja.

## 1.2 Permasalahan

Remotely Operated Vehicle (ROV) secara essensial merupakan sebuah robot bawah air yang mengikuti operator kapal dalam mempertahankan





kenyamanan lingkungan sementara ROV bekerja didalam bahaya lingkungan bawah laut. Sehingga diperlukan adanya kerja kontroller/perancangan sistem pengendalian yang mampu menjaga kestabilan gerak ROV dalam mendukung aktifitas kerja bawah laut.

Dengan melihat permasalahan dasar pada perancangan sistem pengendalian yang ada, baik yang konvensional maupun modern yang telah diuraikan dalam pendahuluan dan semakin berkembangnya sistem pengendalian berbasis intelligent control, maka kebutuhan pengendalian yang dapat merepresentasikan cara berpikir manusia yang dinyatakan dengan variabel linguistik semakin diperlukan.

Untuk itu, inti permasalahan yang ingin dikemukakan pada dasarnya mengusahakan sistem pengendalian otomatis lain sebagai alternatif sistem pengendalian yang ada pada sistem olah gerak ROV sehingga diharapkan dapat memberikan keefektifan/kestabilan gerak pada ROV dalam mengatasi perubahan kondisi/bahaya lingkungan dimana ROV tersebut beroperasi.

Untuk merealisasikan sistem Kendali Logika Fuzzy (KLF) dan menghindari pembahasan yang terlalu meluas pada tugas akhir ini, maka perlu diadakan pembatasan masalah pada sistem olah gerak ROV (DEMON ROV) terutama pada kontrol manuever ROV saat melakukan penjelajahan/sedang bergerak, dengan masukan (berupa sudut posisi, rolling, dan pitching) baik secara manual maupun sensor/otomatis. Selain itu tidak membahas aplikasinya dalam pengaturan besar kecilnya bukaan katup (servovalve).



### **1.3 Metodologi**

Metode – metode yang digunakan untuk mengerjakan tugas akhir ini adalah :

1. Studi literatur

Untuk pengumpulan bahan referensi penunjang yang dapat membantu penulis, baik itu melalui jurnal, paper, buku, maupun website, terutama masalah fuzzy logic dan kontrol sistem olah gerak ROV dalam manuever.

2. Studi lapangan

Mendapatkan data-data teknis untuk memodelkan perancangan yang akan didesain, terutama konfigurasi penempatan thrusters yang berkaitan dengan perancangan sistem kontrolnya sendiri.

3. Perancangan algoritma kontroler/sistem fuzzy

Pada tahap ini dilakukan perancangan dengan program matlab, pertama-tama dilakukan pendefinisian sistem kontrol dan menyusunnya ke dalam tahapan sistem fuzzy.

4. Analisa hasil perancangan.

Pada tahap ini dilakukan analisa mengenai sistem fuzzy yang diterapkan pada sistem kontrol ROV.

### **1.4 Tujuan dan Manfaat**

Tujuan penulisan tugas akhir ini adalah untuk :

Menerapkan/merancang sistem Kendali Logika Fuzzy (KLF) pada sistem olah gerak ROV. Hasil perancangan berupa software akan diterapkan pada sistem pengendalian proses dengan cara simulasi (berupa tampilan dalam bentuk tabel dan grafik).





Sedangkan manfaat penulisan tugas akhir ini adalah untuk :

1. Memberikan alternatif lain sistem pengendalian otomatis dalam pengendalian sistem olah gerak ROV.
2. Diperoleh efektifitas kestabilan gerak ROV dalam mengatasi perubahan kondisi/bahaya lingkungan dimana ROV tersebut beroperasi.

### 1.5 Sistematika Penulisan

Dalam penyusunan laporan tugas akhir ini, dapat tersusun sebagai berikut :

- Bab I Menguraikan pendahuluan yang berisi latar belakang, permasalahan, tujuan dan manfaat, metodologi, dan sistematika penulisan.
- Bab II Menguraikan dasar teori tentang logika fuzzy dan ROV sebagai landasan teori yang akan digunakan dalam penyusunan tugas akhir.
- Bab III Menguraikan pemodelan sistem thruster ROV terutama pada peletakan konfigurasi thruster yang digunakan.
- Bab IV Menguraikan perancangan algoritma controller logika fuzzy terhadap pemodelan thruster yang ada.
- Bab V Menguraikan analisa hasil perancangan berdasarkan hasil simulasi yang tersaji dalam bentuk tabel dan grafik.
- Bab VI Menguraikan kesimpulan dan saran mengenai kemungkinan pengembangan perancangan yang telah dilakukan.



## BAB II

### DASAR TEORI

#### 2.1 Logika Fuzzy

Pada dasarnya manusia mengenal obyek dengan memberikan klasifikasi secara kualitatif, seperti besar, kecil, tinggi, rendah, dsb, yang diklasifikasikan sebagai terminologi linguistik (*Huda, 1997*). Batas antara satu kebenaran dengan kebenaran lainnya tidak tegas dan sering mengandung unsur ketidakpastian. Kebenaran ini disebut kebenaran fuzzy.

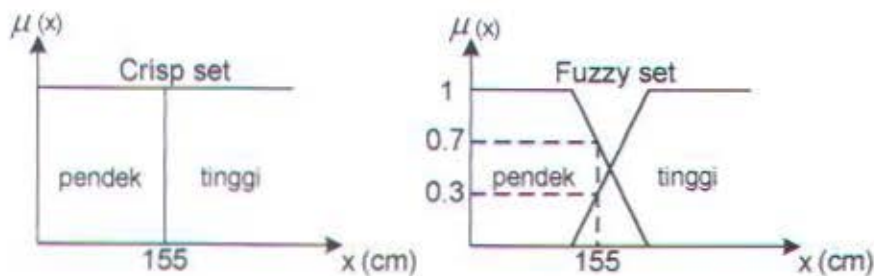
Dengan mengatakan udara cerah bukan berarti bahwa tidak ada sedikitpun awan dilangit. Akan tetapi kita pun tidak bisa mengatakan bahwa udara cerah pada saat langit tertutup awan lebih dari 50 %. Adalah wajar jika dikatakan bahwa udara cerah pada saat awan menutupi sekitar 10/20 %. Masalahnya adalah bagaimana batas pemisah harus ditarik. Jika batas pemisah tersebut berada pada 25 %, apakah berarti 26 % sudah tidak cerah ?. Hal ini tidak bisa diterima oleh akal sehat, karena perbedaan % praktis tidak tampak. Kita dapat menambahkan kualifikasi bahwa 1 % perbedaan (kurang/lain) masih dipandang sebagai cerah. Akan tetapi definisi pada akhirnya akan membawa kita untuk menerima semua tingkatan jumlah awan sebagai cerah, terlepas dari bagaimana cuaca itu sebenarnya. Untuk membatasi kontradiksi ini, istilah cerah dapat disertai oleh vagueness (ketidakjelasan) dengan memberikan perubahan transisi secara bertahap pada derajat jumlah awan dari keadaan cerah menjadi keadaan tidak cerah. Hal inilah yang melahirkan konsep himpunan fuzzy.





Konsep himpunan fuzzy pertama kali diperkenalkan oleh L.A. Zadeh pada tahun 1965 (*Pedrycz, 1996*). Konsep ini kemudian berkembang menjadi teori himpunan fuzzy yang secara luas diterapkan diberbagai bidang dan menjadi salah satu pendekatan yang penting untuk pertimbangan praktis engineers (*Tomsovic, 2000*).

Teori fuzzy dikembangkan untuk mengatasi kelemahan teori “two valued logic” (*Huda, 1997*). Teori ini membagi suatu keadaan hanya menjadi dua kemungkinan saja, misal benar/salah, hitam/putih, dan tidak memberikan kemungkinan suatu nilai yang lain. Teori “two valued logic” yang kemudian diimplementasikan dalam metode pemecahan masalah ternyata sangat efektif dan berhasil sebatas permasalahan dapat deskripsikan secara tepat kuantitasnya. Dalam kenyataannya, suatu permasalahan sering tidak dapat dideskripsikan secara jelas kuantitasnya.



Gb. 2.1 Perbandingan metode konvensional dan fuzzy

Sebagai contoh, orang dengan tinggi 160 cm sulit ditentukan apakah orang ini termasuk kategori tinggi/pendek. Contoh cara pemecahan dengan “two valued logic” adalah dengan memberikan range 0-155 cm adalah pendek dan 155 cm lebih termasuk kategori tinggi. Kesulitan pertama adalah menentukan batas pendek dan tinggi secara kuantitatif karena setiap orang mempunyai nilai yang berbeda. Walaupun dapat maka kesulitan yang kedua adalah menentukan kategori



orang yang mempunyai nilai tinggi 155 cm, yang berarti terletak pada perbatasan tinggi dan pendek. Logika fuzzy membagi satu keadaan dalam interval  $[0,1]$  yang secara intuitif dapat dinyatakan dengan contoh, seperti orang itu pendek, sangat pendek, agak pendek, tinggi, agak tinggi, dst. Jika logika ini diterapkan pada permasalahan diatas maka dapat dikatakan misal, orang dengan tinggi 155 cm mempunyai nilai 0.7 pendek dan 0.3 tinggi, yang berarti orang tersebut cenderung pendek, seperti yang diperlihatkan pada gambar 2.1. Pemecahan seperti ini lebih dapat diterima jika dibandingkan dengan logika konvensional.

Logika fuzzy didasarkan pada bentuk aturan-aturan "Jika.....Maka" (*Negnevitsky*) yang mengkonversikan input ke output, dimana satu himpunan fuzzy akan termasuk ke himpunan fuzzy lainnya. Pada prinsipnya himpunan fuzzy tidak lain adalah perluasan himpunan crisp, yaitu himpunan yang membagi sekelompok individu kedalam dua kategori, yaitu anggota dan bukan anggota, dengan nilai keanggotaanya terletak antara 0 dan 1. Banyak himpunan yang justru lebih cenderung mengikuti karakteristik himpunan fuzzy, misal mobil mahal, rumah mewah, dsb.

Algoritma kontrol berbasis teori himpunan fuzzy merupakan topik yang telah banyak memberikan hasil pada tahun-tahun terakhir ini (*Tomsovic, 2000*). Karakteristik yang paling menarik dari sistem kontrol fuzzy ini adalah kemampuannya untuk mengemulasi perilaku manusia (sebagai operator), yang lebih mendasarkan pada kriteria kualitatif daripada kuantitatif, dan tidak memerlukan model matematik sistem yang rumit.





### 2.1.1 Pandangan Mengenai Logika Fuzzy

Logika Fuzzy merupakan semua tentang relatif pentingnya ketelitian (Zadeh, 1995). Seberapa penting hal itu menjadi benar-benar tepat ketika jawaban dasar diberikan ?. Semua buku tentang logika fuzzy memulai dengan beberapa bagian tentang topik ini. Disini beberapa pandangan mengenai logika fuzzy menurut orang-orang pandai yang dikatakan waktu lalu :

- *Henri Matisse*

Ketelitian bukan kebenaran.

- *Kene Dubos*

Kadang-kadang yang lebih dapat diukur menghasilkan keluaran yang paling penting.

- *Charles Sanders Peirce*

Ketidakjelasan lebih dekat dikerjakan dengan dunia logika daripada friksi mekanik.

- *H.L. Mencken*

Saya percaya bahwa kenihilan merupakan kebenaran yang tidak terkondisikan dan karena itu saya tertantang pada beberapa pernyataan kebenaran positif dan beberapa orang yang membuatnya.

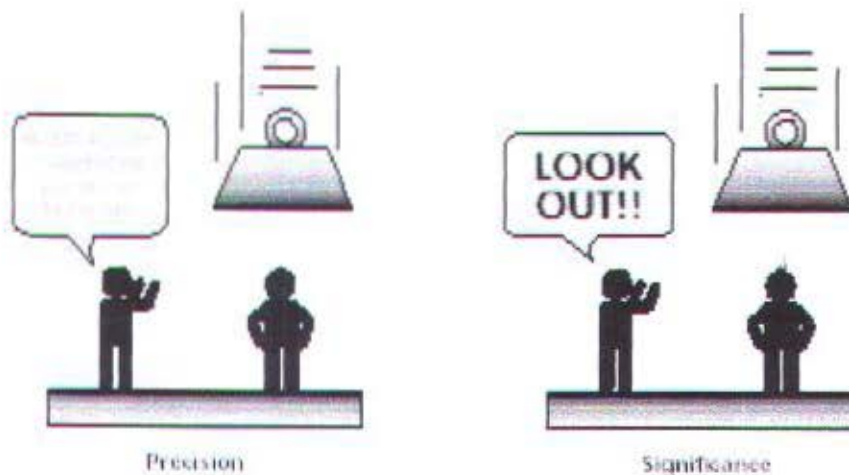
- *Albert Einstein*

Sejauh hukum matematika cenderung real, mereka tidak pasti dan sejauh mereka pasti, mereka cenderung tidak real.

- *Lotfi Zadeh*

Karena kompleksitas muncul, pernyataan ketelitian kehilangan maknanya dan makna pernyataan kehilangan ketelitian.

Logika fuzzy merupakan daerah penelitian yang mengagumkan karena bertugas baik memisahkan antara ketelitian dan ketepatan – sesuatu yang manusia telah perlakukan untuk waktu yang sangat lama. Hal ini dapat diilustrasikan pada gambar 2.2.

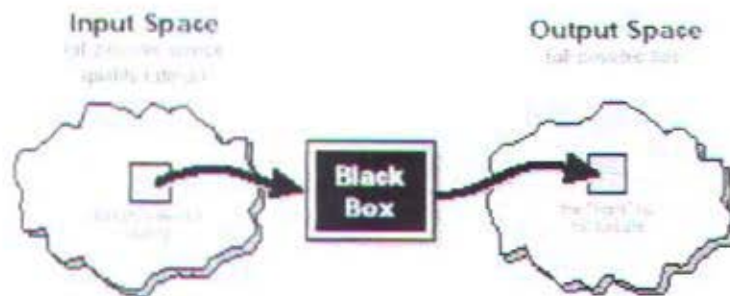


Gb. 2.2 Perbedaan ketelitian dan ketepatan dalam dunia nyata

Logika fuzzy kadang-kadang tampak luar biasa atau menakutkan pada ketidakbiasaan dengan logika fuzzy, tetapi sekali kita membiasakan dengan logika fuzzy, logika fuzzy tampak hampir mengejutkan.

logika fuzzy merupakan cara yang tepat memetakan daerah masukan ke daerah keluaran (Zadeh, 1995). Apakah yang dimaksud dengan memetakan daerah masukan ke daeran keluaran ?. Disini diberikan beberapa contoh : Kamu memberitahukanku, “Seberapa bagus pelayananmu pada restoran”, dan aku memberitahukanmu, “Berapa tip yang akan kamu berikan”. Kamu memberitahukanku, “Seberapa panas kamu menginginkan air”, dan aku akan menyetel katup keran ke kanan. Kamu memberitahukanku, “Seberapa jauh subjek yang kamu foto”, aku akan memfokuskan lensa ke kamu.

Sebuah contoh grafik dari pemetaan daerah masukan ke daerah keluaran ditunjukkan pada gambar 2.3.



Gb. 2.3 Pemetaan daerah masukan ke daerah keluaran

Ini semua hanya masalah memetakan masukan untuk memprediksi keluaran. Antara masukan dan keluaran kita akan letakkan kotak hitam yang melakukan kerja. Apa yang dapat kita masukkan dalam kotak hitam tersebut ?. Sejumlah peralatan seperti : sistem fuzzy, sistem linear, expert system, neural networks, persamaan differensial, tabel interpolasi multi-dimensional, atau bahkan spiritual penasehat, semua itu hanya sebuah nama dari beberapa pilihan yang mungkin. Lepas dari daftar yang ada diatas jika dapat digunakan maka gunakanlah.

Sejumlah cara dibuat untuk kerja kotak hitam, lepas dari itu semua bahwa fuzzy sering merupakan cara terbaik, kenapa harus begitu ?. Lotfi Zadeh, orang yang dianggap sebagai bapak logika fuzzy, berpendapat : “Hampir dalam beberapa kasus, kamu dapat membuat produk yang sama tanpa logika fuzzy, tetapi fuzzy lebih cepat dan murah”.





### 2.1.2 Kelebihan Menggunakan Logika Fuzzy

Disini diberikan beberapa kelebihan menggunakan logika fuzzy :

- ✓ Logika fuzzy secara konsep mudah dipahami

Daripada konsep matematika, penalaran fuzzy sangat mudah. Apa yang membuat fuzzy bagus adalah pendekatan kenaturalannya dan pendekatan dalam kompleksitas.

- ✓ Logika fuzzy fleksibel

Dengan diberikan beberapa sistem, logika fuzzy mudah menyesuaikan/memilah.

- ✓ Logika fuzzy toleran terhadap ketidaktepatan data

Segala sesuatu ketidaktepatan jika kita lihat cukup dekat, sebagian besar peralatan tidak tepat bahkan dalam penelitian yang hati-hati. Penalaran fuzzy membangun pengertian ini kedalam proses daripada merangkaikannya sampai akhir.

- ✓ Logika fuzzy dapat memodelkan fungsi non-linear dari kekompleksitasan

Kita dapat membangun sistem fuzzy untuk menggabungkan beberapa set data masukan ke keluaran.

- ✓ Logika fuzzy dapat dibangun dengan pengalaman seorang ahli

Logika fuzzy mengijinkan kita mengandalkan pengalaman orang yang sudah mengerti sistem dari objek.

- ✓ Logika fuzzy dapat digabung dengan teknik kontrol tradisional

Sistem fuzzy tidak perlu menggantikan metode kontrol tradisional. Dalam banyak kasus, sistem fuzzy memberikan tambahan dan memudahkan penerapan metode kontrol tradisional.





- ✓ Logika fuzzy didasarkan pada bahasa natural

Dasar logika fuzzy adalah komunikasi manusia. Hal ini menopang banyak dari pernyataan lain tentang logika fuzzy.

Pernyataan terakhir mungkin satu yang paling penting dan patut mendapat lebih pembahasan. Bahasa natural yang telah digunakan oleh masyarakat umum setiap harinya sudah terbentuk ribuan tahun dengan sejarah manusia menjadikannya efektif dan tepat. Kalimat yang ditulis dengan bahasa umum memberikan komunikasi efektif. Kita umumnya tidak menyadari hal ini karena bahasa umum/natural, tentunya, sesuatu yang kita gunakan setiap hari. Karena logika fuzzy dibangun atas struktur dari uraian yang bersifat kualitatif dengan menggunakan bahasa sehari-hari/umum, logika fuzzy mudah digunakan.

### 2.1.3 Logika Fuzzy Tidak Digunakan

Logika fuzzy bukan semua penyelesaian. Kapan seharusnya kita tidak menggunakan logika fuzzy ?. Logika fuzzy merupakan cara yang tepat memetakan daerah masukan ke daerah keluaran. Jika kita menemukan adanya ketidaktepatan, cobalah sesuatu yang lain. Jika penyelesaian yang lebih mudah sudah ditemukan, maka gunakanlah. Logika fuzzy merupakan kodifikasi/penyusunan dari pengertian umum – menggunakan pengertian umum ketika kita menerapkannya dan akan memungkinkan membuat keputusan yang benar. Banyak kontroller, misalnya, melakukan pekerjaan dengan baik tanpa menggunakan logika fuzzy. Bagaimana juga, jika kita mengambil waktu yang sama dengan logika fuzzy, kita akan melihatnya dapat lebih bertenaga/powerful untuk penyelesaian secara cepat dan efektif dengan ketidaktepatan dan ke non-linearitasan.



## 2.2 Teori Himpunan Fuzzy

Teori himpunan fuzzy didefinisikan dalam suatu semesta pembicaraan (universe of discourse). Sebuah himpunan fuzzy dapat dipandang sebagai generalisasi dari konsep himpunan klasik (classical set). Pada semesta pembicaraan  $X$ , sebuah himpunan fuzzy ditunjukkan oleh fungsi keanggotaan dimana akan memetakan setiap anggota  $X$  pada rentang keanggotaan antara 0 dan 1

Dibandingkan dengan konsep himpunan klasik yang hanya memetakan setiap anggota  $X$  pada nilai 0 dan 1 (artinya bukan anggota dan anggota penuh), maka konsep teori himpunan fuzzy memberikan keleluasaan pada kita untuk menyatakan kejadian-kejadian dialam nyata (real world) yang mengarah pada kekaburan (fuzziness) dengan tingkat perubahan secara perlahan (gradual). Tingkat perubahan secara perlahan tersebut diungkapkan dengan bahasa natural, dimana secara matematik mempunyai tingkat logika antara 0 dan 1.

### 2.2.1 Himpunan Fuzzy

Suatu himpunan fuzzy  $A$  dalam semesta  $X$  didefinisikan sebagai kumpulan pasangan elemen  $x$  dan fungsi keanggotaan  $\mu_A(x)$ . Fungsi keanggotaan  $\mu_A(x)$  mempunyai nilai dalam interval  $[0,1]$  pada tiap  $x$  dalam  $X$ . Nilai fungsi keanggotaan menunjukkan tingkat keanggotaan elemen  $x$  dalam  $A$ , bila  $\mu_A(x) = 1$  menunjukkan  $x$  merupakan anggota penuh  $A$ , sedang  $\mu_A(x) = 0$  menunjukkan  $x$  bukan anggota  $A$ .

Secara umum himpunan fuzzy dinotasikan sebagai :

$$A = \{(x, \mu_A(x)) / x \in X\} \dots\dots\dots (2.1)$$





Bila  $x$  diskrit dengan  $n$  elemen,  $A$  dapat dinyatakan sebagai :

$$A = \mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \dots + \mu_A(x_n)/x_n$$
$$A = \sum_{i=1}^n \mu_A(x_i)/x_i \quad \dots\dots\dots (2.2)$$

Bila  $x$  kontinyu,  $A$  dapat dinyatakan sebagai :

$$A = \int_x \mu_A(x)/x \quad \dots\dots\dots (2.3)$$

Notasi-notasi diatas, seperti  $(+)$ ,  $(/)$ , bukan operator aritmatik penjumlahan dan pembagian, tetapi menyatakan operator gabungan dan penghubung nilai elemen fuzzy dan tingkat keanggotaannya.

Dalam himpunan fuzzy  $A$ ,  $x$  disebut pendukung (support)  $A$ . Untuk elemen  $x$  dimana  $\mu_A(x) = 0.5$  disebut titik silang (cross-over). Himpunan fuzzy yang mempunyai pendukung tunggal dengan  $\mu_A(x) = 1$  disebut fuzzy tunggal (fuzzy singleton).

### 2.2.2 Fungsi Keanggotaan Himpunan Fuzzy

Fungsi Keanggotaan Himpunan Fuzzy merupakan suatu fungsi untuk menyatakan keanggotaan dari suatu nilai-nilai. Pernyataan tingkat keanggotaan nilai-nilai tersebut dalam semesta pembicaraan himpunan fuzzy tergantung pada fungsi keanggotaan himpunan fuzzy yang digunakan. Secara umum ada dua metode dalam pendefinisian keanggotaan himpunan fuzzy yaitu pendekatan secara numerik dan fungsional.

Secara numerik sebuah nilai yang dinyatakan tingkat keanggotaannya dalam semesta pembicaraan himpunan fuzzy adalah sebagai vektor nilai yang besarnya tergantung level nilai pada pendukung diskrit. Selain itu dapat juga





menggunakan nilai keanggotaan dari bentuk fungsi yaitu dengan mengambil nilai bentuk fungsi untuk tiap pendukung  $x$  yang berhingga jumlahnya. Misalnya :

$$X = \{3, 4, 5, 6, 7\}$$

$$\mu_A(x) = \{0,4; 0,9; 1; 0,9; 0,4\}$$

maka himpunan fuzzy yang didefinisikan untuk himpunan elemen  $x$  diatas adalah sebagai berikut :

$$A = \{ 0,4/3; 0,9/4; 1/5; 0,9/6; 0,4/7\}$$

Sedangkan secara fungsional tingkat keanggotaan sebuah nilai dalam semesta pembicaraan himpunan fuzzy dinyatakan dalam bentuk fungsi keanggotaan dengan pendukung kontinyu. Ada beberapa fungsi yang dapat digunakan dalam himpunan fuzzy, diantaranya :

➤ Fungsi eksponensial

$$\mu_A(x) = \exp\left(-\frac{(x-u)^2}{2\sigma^2}\right) \dots\dots\dots(2.4)$$

➤ Fungsi segitiga

$$\mu(x;a,b,c) = \left\{ \begin{array}{ll} 0 & ; x < a \\ \frac{(x-a)}{(b-a)} & ; a \leq x \leq b \\ \frac{(c-x)}{(c-b)} & ; b \leq x \leq c \\ 0 & ; x > c \end{array} \right\} \dots\dots\dots(2.5)$$

atau

$$\mu_A(x) = 1 - \frac{\sqrt{(x-a)^2}}{b} \dots\dots\dots(2.6)$$

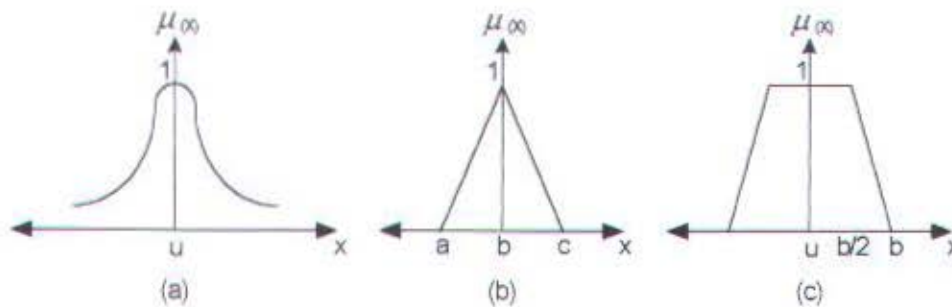




## ➤ Fungsi trapesium

$$\mu_A(x) = \begin{cases} 1 & ; 0 \leq (x-a) \leq \frac{b}{a} \\ 2 - \frac{2\sqrt{(x-a)^2}}{b} & ; \frac{b}{2} < (x-a) \leq b \end{cases} \dots\dots\dots(2.7)$$

Gambar dari bentuk-bentuk fungsi diatas ditunjukkan oleh gambar 2.4.



Gb. 2.4 Bentuk-bentuk fungsi keanggotaan (*Huda, 1997*)

(a) eksponensial, (b) segitiga, (c) trapesium

Dari beberapa bentuk fungsi keanggotaan dalam teori himpunan fuzzy yang sering digunakan adalah bentuk segitiga (triangular shape).

### 2.2.3 Variabel Linguistik

Manusia menyatakan informasi-informasi yang diinderanya dengan ungkapan yang bersifat kualitatif. Ungkapan-ungkapan ini terus menyertai proses berpikir manusia sampai diperoleh pengambilan keputusan yang bersifat kualitatif pula. Sebagai contoh, seorang operator yang sedang mengatur suatu proses secara manual akan menggunakan ungkapan-ungkapan seperti kurang panas, terlalu panas, sedang, dsb.

Dalam logika fuzzy, untuk merepresentasikan cara berpikir manusia diperlukan variabel linguistik sebagai ganti variabel numerik yang biasa



digunakan dalam pendekatan kuantitatif. Variabel linguistik ini berupa ungkapan yang bersifat kualitatif dan berfungsi untuk menyatakan himpunan fuzzy.

Dalam bentuk sederhana variabel linguistik dapat dinyatakan dengan pasangan  $(u, T(u), X)$ ; dimana  $u$  menunjukkan nama variabel dan  $T(u)$  adalah istilah yang menyatakan himpunan  $u$  yaitu seperangkat nama-nama nilai linguistik himpunan fuzzy atas semesta  $X$ . Sebagai contoh, misal variabel suhu akan dinyatakan dengan variabel linguistik, maka perangkat istilah  $T(\text{suhu})$  dapat dituliskan :

$$T(\text{suhu}) = \{\text{rendah, sedang}\}$$

Misal variabel linguistik didefinisikan untuk semesta  $X = [0, 100]$ .

Dalam hal ini, rendah terletak disekitar  $25^{\circ}\text{C}$  dan sedang disekitar  $50^{\circ}\text{C}$ . Bila digunakan fungsi segitiga untuk mendefinisikannya secara fungsioanal maka diperoleh :

$$\text{rendah : } \mu_A(x) = 1 - \frac{\sqrt{(x-25)^2}}{b}$$

$$\text{sedang : } \mu_A(x) = 1 - \frac{\sqrt{(x-50)^2}}{b}$$

dimana  $x \in X$ , harga  $b$  dipilih sedemikian rupa sehingga titik silang terletak di titik  $37.5^{\circ}\text{C}$ , yaitu pendukung dengan nilai keanggoataan  $\mu(37.5) = 0.5$ , baik pada himpunan rendah maupun sedang. Penafsiran secara grafik ditunjukkan seperti pada gambar 2.5.



Gb. 2.5 Penafsiran grafik variabel linguistik



Pemilihan variabel linguistik sebagai aksi pengendalian memegang peranan sangat penting dalam menentukan performansi unit pengendali logika fuzzy atau dengan kata lain kualitas dari pengendali logika fuzzy tergantung dari kualitas rule (*Kosko, 1997*) yang tersusun atas variabel linguistik (*Negnevitsky*). Pada kasus sistem pengendalian, setiap variabel masukan dan keluaran digolongkan dalam variabel linguistik seperti : NB, NS, NK, NL, PK, PS, dan PB.

#### 2.2.4 Operasi Dasar Himpunan Fuzzy

Operasi dasar himpunan fuzzy yang digunakan dalam strategi unit strategi pengendali logika fuzzy terdiri dari operator komplemen, gabungan, dan irisan (*Pedrycz, 1996*). Tinjau variabel A dan B sebagai himpunan fuzzy dalam semesta pembicaraan X dengan fungsi keanggotaan  $\mu_A(x)$  dan  $\mu_B(x)$ , maka peninjauan operasi dasar himpunan fuzzy yang dinyatakan dengan operator komplemen, gabungan, dan irisan adalah :

- Komplemen

Komplemen himpunan fuzzy A atau B dengan masing-masing fungsi keanggotaan  $\mu_A(x)$  atau  $\mu_B(x)$ , dinyatakan dengan :

$$\bar{A} = 1 - \mu_A(x); x \in X \dots\dots\dots (2.8)$$

- Gabungan

Gabungan 2 himpunan fuzzy A dan B dengan masing-masing fungsi keanggotaan  $\mu_A(x)$  dan  $\mu_B(x)$ , dinyatakan dengan :

$$\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}; x \in X \dots\dots\dots (2.9)$$





- Irisan

Irisan 2 himpunan fuzzy A dan B dengan masing-masing fungsi keanggotaan  $\mu_A(x)$  dan  $\mu_B(x)$ , dinyatakan dengan :

$$\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}; x \in X \dots\dots\dots(2.10)$$

### 2.3 Struktur Dasar Unit Kendali Logika Fuzzy

Perancangan kendali logika fuzzy merupakan gabungan aspek pendefinisian himpunan fuzzy dengan aspek logika fuzzy. Kendali logika fuzzy mampu mengakomodasi informasi numerik dari sistem pengukuran dan informasi linguistik yang diperoleh dari diskripsi operator yang berpengalaman. Dengan kedua informasi ini memungkinkan efektifitas pendekatan secara engineering sense dalam menyelesaikan suatu masalah sistem pengendalian.

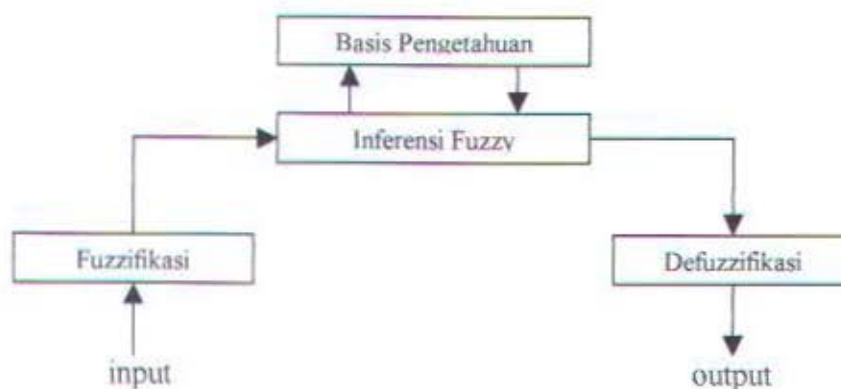
Kendali logika fuzzy dengan menggunakan basis pengetahuan dan ungkapan linguistik mampu merepresentasikan cara kerja seorang operator yang berpengalaman. Aturan pada kendali logika fuzzy adalah himpunan aturan-aturan kontrol linguistik yang diturunkan secara heuristik berdasarkan karakteristik proses dan pengalaman operator. Dengan kata lain kendali logika fuzzy merupakan suatu kontroler yang pada prinsipnya mengubah strategi kontrol linguistik kedalam strategi kontrol otomatis.

Secara umum kendali logika fuzzy dapat dipandang sebagai sistem kontrol loop tertutup yang memiliki 4 bagian pokok seperti yang ditunjukkan oleh gambar 2.6. Keempat bagian tersebut yaitu : unit fuzzifikasi, unit basis pengetahuan yang terdiri dari basis data dan basis aturan, unit inferensi fuzzy, dan unit defuzzifikasi.



Keempat bagian tersebut mempunyai fungsi sebagai berikut :

- Fuzzifier; berfungsi untuk mentransformasikan sinyal masukan yang bersifat crisp (bukan fuzzy) ke himpunan fuzzy dengan menggunakan operator fuzzifier.
- Basis pengetahuan; yang mendefinisikan himpunan fuzzy atas daerah-daerah masukan dan keluaran dan menyusunnya dalam perangkat aturan kontrol.
- Inferensi fuzzy; merupakan inti dari kendali logika fuzzy yang mempunyai kemampuan seperti manusia dalam mengambil keputusan. Unit ini akan mendefinisikan besar derajat keanggotaan dan variabel linguistik dari aksi pengendalian yang dilakukan untuk masing-masing aturan kontrol berdasarkan fungsi implikasi yang digunakan.
- Defuzzifier; berfungsi untuk mentransformasikan kesimpulan yang bersifat fuzzy menjadi sinyal yang sebenarnya yang bersifat crisp dengan menggunakan operator defuzzifier.



Gb. 2.6 Struktur dasar kendali logika fuzzy (Huda, 1997)

### 2.3.1 Fuzzifikasi

Unit fuzzifikasi berfungsi untuk mengubah data crisp keluaran dari plant yang terukur oleh sensor menjadi data fuzzy yaitu berapa derajat keanggotaan dan





variabel linguistik. Pada unit fuzzifikasi juga dikenal bentuk fungsi keanggotaan yaitu fungsi yang menyatakan tingkat keanggotaan data crisp tertentu terhadap data fuzzy tertentu. Ada berbagai macam bentuk fungsi keanggotaan, seperti bentuk trapesoidal, triangular, gaussian, dll. Pada penulisan TA ini bentuk fungsi keanggotaan yang digunakan adalah bentuk segitiga, seperti pada gambar 2.4(b) dengan menggunakan persamaan 2.5.

Dengan harga derajat keanggotaan masukan  $x$  berada pada nilai 0 sampai 1, misal  $\mu_A(x) = 0.6$ , berarti bahwa data crisp  $x$  memiliki derajat keanggotaan sebesar 0.6 untuk variabel linguistik.

### 2.3.2 Inferensi Fuzzy (Fuzzy Inference Rules)

Pada umumnya aturan fuzzy dinyatakan dalam bentuk logika “Jika.....Maka” yang merupakan dasar relasi fuzzy (*Brown, 1994*). Relasi fuzzy  $R$ , dalam basis pengetahuan fuzzy didefinisikan sebagai implikasi fuzzy (fuzzy implication).

Dalam penalaran logika fuzzy, terdapat dua metode dalam mengungkapkan aturan inferensi fuzzy, yaitu : Generalized Modus Ponens (GMP) dan Generalized Modus Tollens (GMT) (*Pedrycz, 1996*). Skema GMP disebut juga sebagai teknik penalaran langsung (direct reasoning) menggunakan hukum pemisahan (law of detachment), sedangkan skema GMT disebut juga teknik penalaran tidak langsung (indirect reasoning) menggunakan hukum kontraposisi.

Tinjau variabel  $A$ ,  $A'$ ,  $B$ ,  $B'$ , sebagai himpunan fuzzy dan  $x$ ,  $y$ , sebagai variabel linguistik, maka GMP dan GMT dapat dinyatakan sebagai :





- Generalized Modus Ponens

Premis 1 (pengetahuan) : jika x adalah A, maka y adalah B

Premis 2 (kenyataan) : x adalah A'

-----

Konsekuensi (kesimpulan) : y adalah B'

Pada kasus ini, konsekuensi B' dapat dinyatakan dengan :

$$B' = A' \circ R$$

dimana R menyatakan relasi fuzzy dari implikasi fuzzy "Jika.....Maka", o adalah operator komposisional.

Inferensi fuzzy adalah metode pengambilan keputusan yang disebut sebagai common sense inference, karena hal itu dilakukan sama dengan manusia biasa, seperti sebagai berikut :

Pengetahuan : jika air sangat panas maka tambahkan air dingin  
secukupnya

Fakta : air cukup panas

-----

Kesimpulan : tambahkan sedikit air dingin

Tampak dari contoh diatas bahwa inferensi fuzzy merepresentasikan aksi yang akan diambil manusia.

- Generalized Modus Tollens

Premis 1 (pengetahuan) : jika x adalah A, maka y adalah B

Premis 2 (kenyataan) : x adalah B'

-----

Konsekuensi (kesimpulan) : y adalah A'



Pada kasus ini, konsekuensi  $A'$  dapat dinyatakan dengan :

$$A' = R_{ij} B'$$

Aturan inferensi GMT disebut juga aturan inferensi arah mundur (backward inference), biasanya digunakan oleh sistem pakar (expert system) khususnya dalam bidang diagnosa medis (medical diagnosis). Aturan inferensi GMP disebut juga aturan inferensi arah maju (forward inference) yang secara luas digunakan dalam penerapan logika fuzzy, khususnya dalam bidang pengendalian.

### 2.3.3 Basis Pengetahuan Fuzzy

Basis pengetahuan fuzzy terdiri dari beberapa aturan fuzzy (fuzzy rule) yang dikelompokkan dalam suatu basis aturan disebut basis aturan fuzzy (fuzzy rule base).

Basis pengetahuan fuzzy merupakan bentuk perluasan dari aturan inferensi fuzzy dan disesuaikan dengan struktur unit pengendali yang dirancang. Secara keseluruhan bentuk basis pengetahuan fuzzy adalah :

Aturan 1 : jika  $x$  adalah  $A_1$  dan  $y$  adalah  $B_1$  maka  $Z$  adalah  $C_1$

Aturan 2 : jika  $x$  adalah  $A_2$  dan  $y$  adalah  $B_2$  maka  $Z$  adalah  $C_2$

Aturan 3 : jika  $x$  adalah  $A_3$  dan  $y$  adalah  $B_3$  maka  $Z$  adalah  $C_3$

⋮

Aturan  $N$  : jika  $x$  adalah  $A_N$  dan  $y$  adalah  $B_N$  maka  $Z$  adalah  $C_N$

dimana  $A_N$ ,  $B_N$ ,  $C_N$ , adalah nilai linguistik dari variabel  $x$ ,  $y$ ,  $z$ , pada masing-masing semesta pembicaraan (universe of discourse)  $X$ ,  $Y$ ,  $Z$ .



### 2.3.4 Penurunan Aturan Kontrol Fuzzy

Salah satu cara menurunkan aturan kontrol fuzzy adalah dengan mengumpulkan aturan-aturan kontrol yang dibentuk dari analisa perilaku obyek atur secara heuristik. Aturan kontrol diturunkan dengan jalan mengoreksi simpangan keluaran plant dari keadaan yang diinginkan. Penurunan benar-benar dilakukan secara heuristik dengan mengandalkan pengetahuan kualitatif. Hasil yang diperoleh akan berupa prototype aturan kontrol secara garis besar sehingga hasilnya masih perlu disempurnakan.

Ada beberapa metode yang dapat digunakan untuk menyempurnakan prototype aturan kontrol itu. King dan Mamdani, mengusulkan metode penyempurnaan aturan kontrol dengan cara yang disebut pemetaan skala (scale mappings) (Huda, 1997). Prinsipnya adalah menentukan sinyal atur sedemikian rupa sehingga trayektori sistem loop tertutup berakhir di keadaan yang diinginkan.

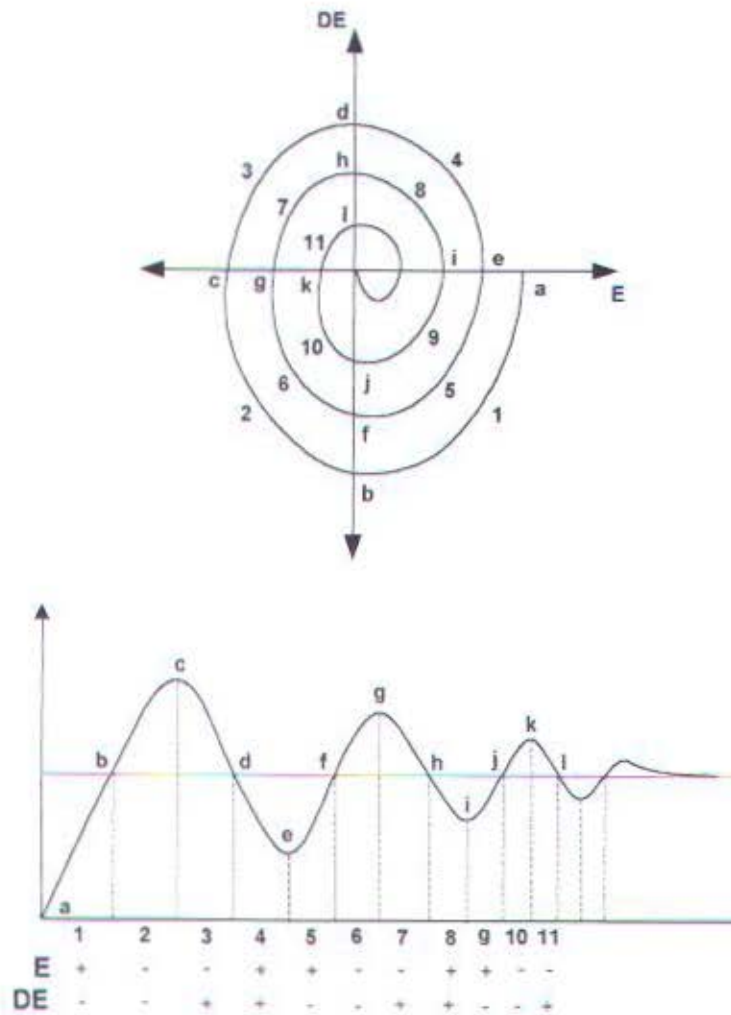
Gambar 2.7 menunjukkan tanggapan plant loop tertutup terhadap variabel masukan plant, yaitu error ( $e$ ) dan perubahan error ( $\Delta e$ ). Untuk memperjelas trayektori dapat digunakan grafik yang menyatakan hubungan antara error ( $e$ ) dan perubahan error ( $\Delta e$ ), sedangkan variabel keluaran kendali logika fuzzy berupa perubahan sinyal atur ( $\Delta u$ )/sinyal atur ( $u$ ).

Pada penurunan prototype ini pembagian ruang masukan adalah 3, yang berarti ada 3 nilai linguistik yaitu positif (P), negatif (N), dan nol (NL). Peninjauan kualitatif per titik (bertanda huruf) pada grafik memberikan penalaran yang menghasilkan prototype aturan kontrol pada tabel 2.1. Sedangkan peninjauan secara kualitatif per daerah (bertanda angka) pada grafik tanggapan maupun pada bidang fasa menunjukkan bahwa daerah 1 mempunyai pengaruh pada





pemendekkan waktu naik (rise time), dan daerah 2 berhubungan dengan pengurangan lewatan (overshoot), secara heuristik penalaran ini memberikan penambahan/penyempurnaan aturan kontrol seperti pada tabel 2.2.



Gb. 2.7 Tanggapan sistem loop tertutup

Aturan No.	e	$\Delta e$	u	titik referensi
1	P	NL	P	a,e,i
2	NL	N	N	b,f,j
3	N	NL	N	c,g,k
4	NL	P	P	d,h,l
5	NL	NL	NL	set point

Tab. 2.1 Prototipe aturan kontrol linguistik dengan 3 nilai



Aturan No.	e	$\Delta e$	u	titik referensi
6	P	N	P	1(rise time),5
7	N	N	N	2(overshoot),6
8	N	P	N	3,7
9	P	P	P	4,8
10	P	N	NL	9
11	N	P	NL	10

Tab. 2.2 Penyempurnaan aturan kontrol dengan 3 nilai

Kinerja yang lebih baik dapat diperoleh dengan pembagian ruang yang lebih banyak, misal 7 nilai yaitu negatif besar (NB), negatif sedang (NS), negatif kecil (NK), nol (NL), positif kecil (PK), positif sedang (PS), positif besar (PB). Pembagian ruang dengan 7 nilai sering digunakan pada perancangan kendali logika fuzzy, karena pembagian ruang tersebut tidak dapat mewakili ruang keseluruhan dan didapat hasil yang baik.

Prototipe dan penyempurnaan berturut-turut ditunjukkan oleh tabel 2.3 dan 2.4. Secara langsung pembagian ruang masukan mempengaruhi jumlah aturan kontrol yang dihasilkan. Pada contoh sistem ini, jumlah aturan kontrol yang dihasilkan adalah  $7 \times 7$  aturan kontrol.

Hasil yang didapat dari penyempurnaan aturan kontrol tersebut dapat dibuat aturan kontrol yang lengkap. Mac Vicar – Whelan yang meneliti masalah penyempurnaan secara lengkap aturan kontrol ini memberikan tabel keputusan sinyal atur yang disebut tabel keputusan Mac Vicar – Whelan. Tabel 2.5 menunjukkan aturan kontrol fuzzy secara lengkap.



Aturan No.	e	$\Delta e$	u	titik referensi
1	PB	NL	PB	a
2	PS	NL	PS	e
3	PK	NL	PK	i
4	NL	NB	NB	b
5	NL	NS	NS	f
6	NL	NK	NK	j
7	NB	NL	NB	c
8	NS	NL	NS	g
9	NK	NL	NK	k
10	NL	PB	PB	d
11	NL	PS	PS	h
12	NL	PK	PK	l
13	NL	NL	NL	set point

Tab 2.3 Prototipe aturan kontrol dengan 7 nilai

Aturan No.	e	$\Delta e$	u	titik referensi
14	PB	NK	PS	1
15	PK	NB	NS	1
16	NB	PS	NS	3
17	NS	PB	PS	3
18	PK	NK	NL	9
19	NK	PK	NL	11

Tab. 2.4 Penyempurnaan aturan kontrol dengan 7 nilai





error	perubahan error							
		NB	NS	NK	NL	PK	PS	PB
	NB	NB	NB	NB	NB	NS	NK	NL
	NS	NB	NB	NB	NS	NK	NL	PK
	NK	NB	NB	NS	NK	NL	PK	PS
	NL	NB	NS	NK	NL	PK	PS	PB
	PK	NS	NK	NL	PK	PS	PB	PB
	PS	NK	NL	PK	PS	PB	PB	PB
	PB	NL	PK	PS	PB	PB	PB	PB

Tab. 2.5 Keputusan aturan kontrol fuzzy Mac Vicar – Whelan (*Huda, 1997*)

Tabel ini memberikan jumlah aturan kontrol secara maksimum. Pola ini berlaku bagi kendali logika fuzzy yang menggunakan masukan error dan perubahan error. Modifikasi pada tabel ini untuk memperbaiki kinerja dapat dilakukan pada batas-batas tertentu.

### 2.3.5 Teknik Penalaran Fuzzy (Fuzzy Reasoning)

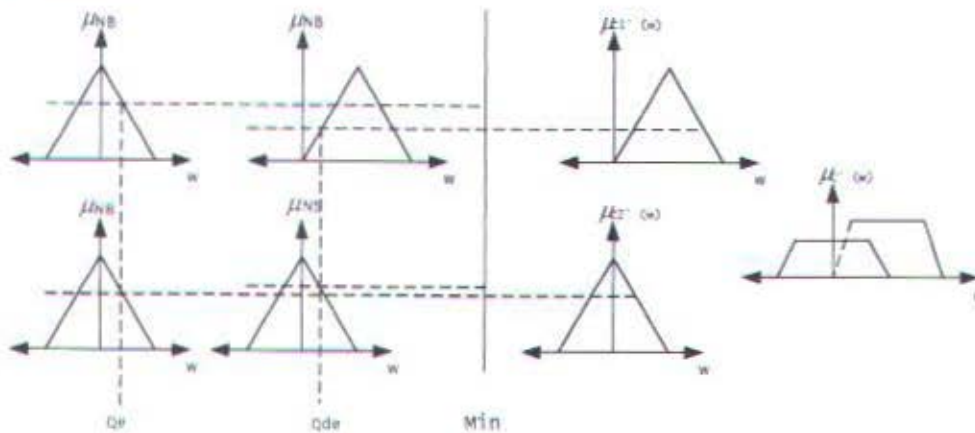
Pada teknik penalaran fuzzy, terjadi pemberlakuan bahwa masukan pengukuran sistem (masukan unit pengendali logika fuzzy) dalam domain crisp diperlakukan fungsi tunggal (fuzzy singleton). Tinjau variabel  $A' = u_0$  dan  $B' = v_0$ , maka dengan aturan inferensi Minimum – Mamdani Rc diperoleh :

$$R_c = \alpha_i \wedge \mu_{ci}(z) \dots\dots\dots(2.11)$$

dimana  $\alpha_i = \mu_{Ai}(u_0) \wedge \mu_{Bi}(v_0)$

$\alpha_i$  menyatakan faktor bobot yang menunjukkan nilai kontribusi aksi pengendalian fuzzy pada aturan ke-i,  $\mu_{Ai}(u_0)$  dan  $\mu_{Bi}(v_0)$  masing-masing menyatakan nilai keanggotaan masukan pengukuran pada semesta pembicaraan  $u$  dan  $v$ . Secara

grafik teknik penalaran menggunakan inferensi Minimum – Mamdani ditunjukkan dalam Gambar 2.8.



Gb. 2.8 Aturan Inferensi Minimum – Mamdani

Dari aturan inferensi Minimum – Mamdani yang diterapkan, maka fungsi keanggotaan dari aksi pengendalian dinyatakan dengan :

$$\mu_{C'}(z) = \bigcup_{i=1}^n \alpha_i \wedge \mu_{C_i}(z) \dots\dots\dots(2.12)$$

dimana  $\alpha_i$  adalah himpunan fuzzy aksi pengendalian. Fungsi keanggotaan yang didapatkan dari persamaan 2.10 adalah fungsi keanggoan dari aksi pengendalian. Dengan fungsi keanggotaan yang didapat dari tiap-tiap aturan, maka akan diperoleh nilai keanggotaan fuzzy ( $\mu_{C'}$ ) yang merupakan representasi terbaik keluaran unit pengendali logika fuzzy. Keluaran unit pengendali logika fuzzy tersebut terlebih dahulu harus diubah dalam domain variabel non-fuzzy (crisp) melalui proses defuzzifikasi.

### 2.3.6 Defuzzifikasi

Pada dasarnya defuzzifikasi adalah pemetaan ruang aksi pengendali (domain fuzzy) kedalam ruang aksi pengendali non-fuzzy (crisp). Secara garis besar strategi defuzzifikasi ditunjukkan untuk menghasilkan aksi pengendalian non-fuzzy yang merupakan representasi terbaik dari kemungkinan aksi pengendali





fuzzy yang tersimpulkan. Secara umum defuzzifikasi dapat dinyatakan dalam bentuk :

$$Z_o = \text{defuzzifier}(z) \dots\dots\dots(2.13)$$

Metode defuzzifikasi yang sering digunakan adalah titik pusat gravitasi (COG = Center of Gravity) yang secara matematik dapat dinyatakan :

$$z_o = \frac{\sum_{i=1}^n \mu_i(z) \times z}{\sum_{i=1}^n \mu_i(z)} \dots\dots\dots(2.14)$$

dimana  $z_o$  adalah aksi pengendali fuzzy dalam domain crisp,  $\mu_i(z)$  adalah nilai keanggotaan (faktor bobot) yang merupakan hasil dari aturan inferensi Minimum – Mamdani,  $z$  adalah nilai keluaran crisp tiap aturan.

#### 2.4 Kontrol Subsea (Subsea Controls)

Sistem kontrol subsea merupakan bagian yang paling kritis dari setiap instalasi subsea (*Goodfellow, 1996*). Pada waktu yang bersamaan, biayanya hanya sering memberikan bagian yang paling minimal dari keseluruhan biaya kontrol subsea secara komplit (umumnya antara 3 – 10 %). Bagaimanapun juga, sejumlah komponen yang menyusun sistem kontrol untuk instalasi utama jauh melebihi dari total seluruh komponen subsea lainnya.

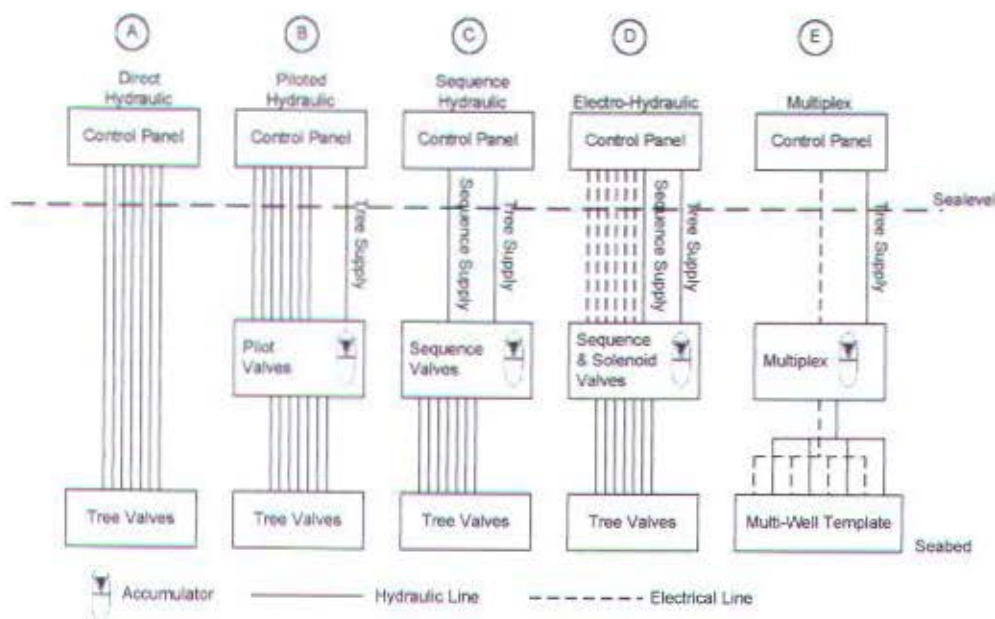
Penempatan peralatan kontrol secara ekstrem sangat penting. Pertimbangan secara hati-hati dapat menyebabkan pengurangan dalam panjang perpipaan, pengabelan, dan sejumlah penghubung perlengkapan instalasi juga perbaikan. Meskipun sistem kontrol sering dikhususkan sebagai peralatan tambahan setelah peralatan cadangan sudah dibangun. Baru-baru ini instalasi sudah menunjukkan bahwa sistem kontrol subsea sudah lebih canggih, keandalannya tinggi, dan memberikan kontrol peralatan subsea beberapa mil jauhnya dengan keyakinan dan kemudahan.





Sistem kontrol subsea, baik yang mudah/ sederhana dengan menggunakan hidrolik atau yang lebih rumit dengan electro-hydraulic, didefinisikan sebagai salah satu komponen aktif subsea (*Goodfellow, 1996*). Sebagian besar dalam hal kontrol ini dengan menggunakan hydraulic power unit (HPU). Saat ini, terdapat 5 tipe sistem kontrol, 3 diantaranya; direct hydraulic, piloted hydraulic, dan sequential hydraulic, semata-mata hanya bertumpu pada gerakan hydraulic fluid untuk mengoperasikan aktuator subsea. 2 tipe lainnya; electro-hydraulic dan multiplexed electro-hydraulic, yang mengkombinasikan sinyal hidrolik dan listrik untuk mengontrol aktuator.

Setiap tipe baru dari sistem yang sudah dikembangkan memberikan kemampuan kontrol. Bagaimanapun, tidak semua sistem kontrol cocok untuk bidang pengembangan. Pemilihan sistem memerlukan spesifikasi yang jelas dari daerah operasioanal, persyaratan sebagai titik awal. Kemudian harus dibandingkan dengan biaya, kemampuan, dan keandalannya sebelum sistem tersebut ditetapkan. Gambar 2.9 menunjukkan 5 alternatif sistem.



Gb. 2.9 Alternatif pemilihan sistem kontrol



#### 2.4.1 Tipe Kontrol Subsea

Disini diberikan beberapa tipe kontrol subsea yang tersedia, diantaranya :

##### A. Direct Hydraulic

Sistem kontrol direct hydraulic secara umum paling mudah dan sebagian besar memberikan keandalan. Setiap subsea function memerlukan hydraulic flowpath dari surface. Kemudian berjalan dari surface menghasilkan pressurized fluid yang dialirkan melalui flowpath ke aktuator hidrolik subsea yang dikehendaki. Sistem ini dengan one line per device, sangat cocok untuk aplikasi dimana jarak kontrol relatif pendek dan dibatasi sejumlah subsea function yang diperlukan.

Karena jarak dari sumber kontrol terlalu panjang atau lama, response time secara drastis meningkat. Karena sejumlah function memerlukan peningkatan kontrol, sehingga membuat kompleksitas dan berpengaruh pada cost dari umbilical. Akibatnya cost dan ukuran umbilical secara cepat meningkat melebihi biaya keseluruhan sistem, dalam beberapa kasus mencapai 50 – 60 % dari biaya total sistem.

##### B. Piloted Hydraulic

Sistem piloted hydraulic meningkatkan response time melebihi sistem direct, dengan menyimpan energi hydraulic pressure pada worksite dengan pilot valve yang menyebabkan aksi. Ukuran umbilical secara kasar sama (meskipun lebih kecil) daripada sistem direct karena setiap katup pengoperasian subsea memerlukan pilot/sinyal line didalam umbilical. Didalam penambahan sinyal line ini, umbilical juga membawa main pressure line.





Aktuator diaktifkan dengan mengirimkan sinyal hidrolik ke pilot valve. Pilot valve terbuka dan membuat fluida mengalir dari supply line dan storage accumulator ke aktuator yang dipilih.

Sistem piloted memberikan time response yang baik. Dengan aspek lain seperti kemampuan monitoring, ukuran umbilical, dan kemampuan menyesuaikan diri pada daerah kerja, mempunyai kesamaan batasan dengan sistem direct hydraulic. Dengan penambahan, ketika dibandingkan dengan sistem direct hydraulic, sistem piloted lebih kompleks dan sedikit handal. Untuk mengurangi masalah kekurangan yang ada seperti ukuran umbilical yang besar, panjang/lamanya jarak kontrol, dan banyaknya sejumlah subsea function, diperkenalkan sistem sequential hydraulic.

#### C. Sequential Hydraulic

Seperti sistem piloted, sistem sequential mempunyai accumulator dan katup subsea, tetapi ukuran umbilical secara signifikan berkurang. Kontrol subsea diperoleh dengan mengirimkan sinyal tekanan ke deretan katup yang mengatur awal pengoperasian pada sinyal tekanan tertentu tersebut. Pada tekanan ini, fluida hidrolik diarahkan ke aktuator yang dipilih.

Sistem juga dibatasi dengan sejumlah kombinasi katup yang dapat dioperasikan, karena besarnya kenaikan tekanan dibatasi dengan batas maksimum keamanan pengoperasian tekanan pada hidrolik dan besarnya kenaikan tekanan diperlukan adanya perbedaan antara kenaikan tekanan yang satu dengan berikutnya.

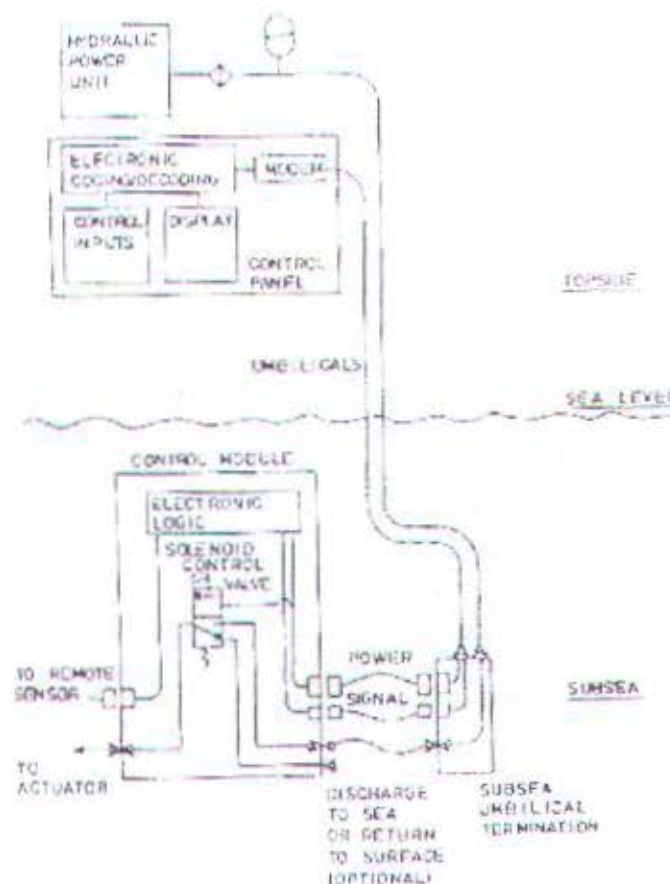




Untuk pengoperasian pada perairan yang lebih dalam, memerlukan peningkatan pada sistem kontrol, salah satunya dengan sistem kontrol electro-hydraulic dan multiplexed electro-hydraulic.

#### D. Electro- Hydraulic

Pada dasarnya, sistem kontrol electro-hydraulic merupakan kontrol piloted hydraulic dengan fungsi piloted diganti dengan sinyal listrik. Setiap aktuator subsea memerlukan sinyal konduktor dari surface, dengan cara yang sama bahwa piloted hydraulic memerlukan sinyal line. Sistem electro-hydraulic sangat bagus penggunaannya dengan hanya dibatasi sejumlah peralatan dari aksi subsea. Tipe sistem kontrol electro-hydraulic ditunjukkan pada gambar 2.10.



Gb. 2.10 Sistem kontrol electro-hydraulic



#### E. Multiplexed Electro- Hydraulic

Sebuah versi pengembangan dari sistem electro-hydraulic dinamakan sistem multiplexed electro-hydraulic. Sistem ini mampu mengontrol sejumlah aktuator, sangat cepat, menggunakan hanya relatif kecil supply umbilical.

Ketika sinyal digital dikirim ke multiplexer, multiplexer mengkoordinasikan ke katup solenoid yang dipilih, karena itu directing hydraulic fluid dari supply umbilical ke aktuator terhubung. Keberadaan dari multiplexer memperkenalkan kemampuan mengontrol tekanan, temperatur, dan posisi katup dengan peralatan sinyal listrik, tanpa adanya kerumitan dalam hubungan listrik melalui umbilical secara operasional, sistem multiplexer menyediakan kemampuan mengontrol yang tidak tersedia pada setiap sistem hidrolik lainnya, kecepatan pengoperasiannya lebih tinggi dari sistem hidrolik lainnya.

### 2.5 ROV

ROV (Remotely Controlled Vehicle) telah digunakan selama 30 tahun terakhir pada seluruh dunia oleh organisasi yang berbeda untuk mendapatkan jalan masuk (access) ke lokasi bawah laut yang tidak mudah dilakukan oleh penyelam/peralatan yang lainnya. Secara histori/sejarah ROV dapat dinamakan remotely controlled cameradan tooling platform. Belakangan ini ROV telah juga melaksanakan tugas yang dapat dilakukan penyelam karena alasan keamanan dan ekonomis. Peningkatan dalam keandalan dan performance dari peralatan sudah dikembangkan 5 tahun terakhir ini dibanding dengan sebelumnya.

Remotely operated vehicle (ROV) secara esensial merupakan sebuah robot bawah air yang mengikuti petunjuk operator kapal dalam mempertahankan





kenyamanan lingkungan sementara ROV bekerja didalam bahaya lingkungan bawah laut (*Michel, 2002*). Didalam aktifitas marine, termasuk penelitian pengetahuan dan eksploitasi komersial sumber kekayaan laut, ROV merupakan salah satu peralatan yang penting. Baru-baru ini kemajuan dalam teori kontrol dan robotik menyebabkan peningkatan kemampuan ROV (*Conte, 2002*).

Sistem total ROV termasuk dari kapal terhubung ke kontrol van dan operator pada surface dengan kabel umbilical, handling system untuk mengontrol dinamik kabel, launch system (seperti pada gambar 2.11), dan terhubung dengan power supplies. Umbilical membawa power, perintah, dan sinyal kontrol ke kapal, kondisi dan sensor data dikembalikan ke operator diatas. Didalam beberapa kasus, umbilical terhitung juga komponen tambahan kekuatan yang memperkenankan pencarian peralatan berat/ kapal karam.



Gb. 2.11 Launch system ROV

ROV dapat bervariasi dalam ukurannya, dari kapal kecil dengan TV untuk pengamatan sederhana sampai sistem kerja kompleks yang dapat mempunyai beberapa manipulator, TV, kamera video, dan perlengkapan lainnya. Saat ini, kemajuan teknologi memperkenankan beberapa ROV yang menumpahkan (shed) kabelnya, sehingga bebas menjelajah lautan tanpa batasan fisik. Sistem baru ini





yang dioperasikan dengan battery dinamakan autonomous underwater vehicles (AUVs) dan digunakan untuk ocean search dan oceanographic research.

### **2.5.1 Sejarah Singkat ROV**

Secara tepat, siapa yang dipercaya dengan pengembangan ROV yang pertama mungkin tetap dipersoalkan, akan tetapi, ada 2 yang patut dipercaya. PUV (Programmed Underwater Vehicle) sebuah torpedo yang dikembangkan oleh Luppis-Whitehead Automobile di Austria pada tahun 1864, akan tetapi, pelopor pertama ROV, dinamakan POODLE, dikembangkan oleh Dimitri Rebikoff pada tahun 1953.

Angkatan laut AS dihargai dalam pengembangan teknologi pada operasional pencarian bawah laut dengan mengembangkan robot untuk pencarian perlengkapan senjata yang hilang selama melakukan percobaan di laut. ROV mendapat popularitas ketika sistem CURV (Cable Controlled Underwater Recovery Vehicle) angkatan laut AS menemukan sebuah bom atom yang hilang di Palomares, Spanyol dalam sebuah kecelakaan aircraft pada tahun 1966, kemudian menyelamatkan pilot kapal selam Sunken di Cork, Irlandia pada tahun 1973, hanya beberapa menit.

Langkah berikutnya dalam pengembangan teknologi dilakukan oleh perusahaan komersial yang melihat ke depan pada ROV mendukung operasional minyak lepas pantai. 2 dari ROV yang pertama dikembangkan untuk kerja offshore adalah RCV-225 dan RCV-150 dikembangkan oleh HydroProduct di AS. Banyak perusahaan lain yang mengembangkan hal yang sama untuk kapal-kapal inspeksi kecil. Sekarang, eksplorasi minyak berpindah dari perairan dalam ke



perairan yang lebih dalam, ROV menjadi bagian yang essential dalam operasional, tidak hanya kemampuannya, tetapi keandalannya yang tinggi.

Dengan ROV bekerja sedalam 10000 ft dalam mendukung minyak lepas pantai dan tugas lainnya, teknologi telah mencapai tingkat biaya yang efektif yang memperkenankan orang dari departement angkatan kepolisian sampai institusi akademik untuk mengoperasikan kapal dari range kapal inspeksi kecil sampai sistem penelitian laut dalam.

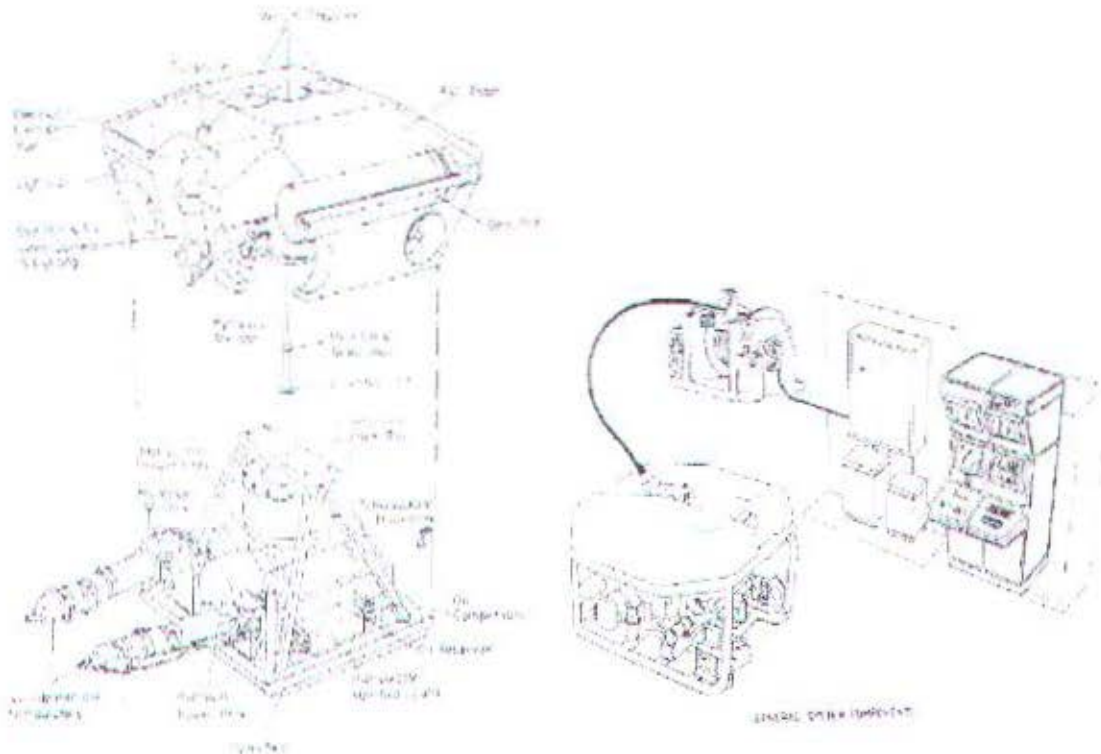
### 2.5.2 Tipe ROV

Industri telah melihat perkembangan dari 3 tipe utama ROV (*Goodfellow, 1996*), diantaranya :

- 1) *Observation vehicles* – membawa TV kamera, mempunyai sedikit kesamaan dalam inspeksi/kerja dalam hubungannya dengan penyelam. Observation vehicle digambarkan dengan adanya lightweight dan mempunyai relatif kecil power thruster. Oleh karena itu, observation vehicle mempunyai kesulitan mempertahankan keadaan dalam setiap arah. Meskipun begitu, observation vehicle merupakan peralatan yang berguna.
- 2) *Specialized ROV* – lebih besar dan bertenaga yang peranan utamanya dalam operasional survey pipeline. Range panambahan peralatan termasuk sub-bottom profilers, side scan sonar, dll yang diberikan kontraktor untuk menyimpan (back up) 2 atau lebih TV kamera yang kapal bawa. Sekali terbuka, bukan tidak umum baginya tetap terselam untuk seminggu atau lebih pada pekerjaannya.
- 3) *Modular multi-task ROV* – sedangkan 2 tipe ROV diatas merupakan model standar, modular multi-task ROV biasanya tujuan pendesignannya untuk



pekerjaan yang spesifik dan dinamakan menurut penyesuaian (seperti pada gambar 2.12).



Gb. 2.12 Remote operated vehicles (sumber : OSEL)

### 2.5.3 Daerah Kerja ROV

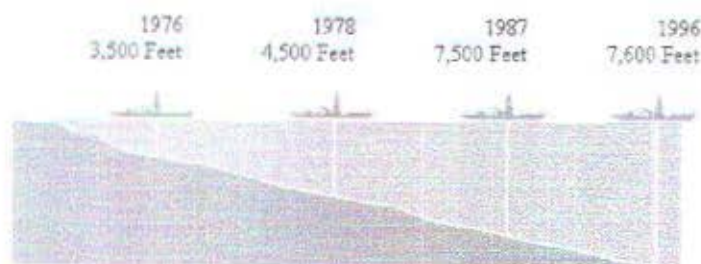
Tugas ROV dalam mendukung pengembangan dan eksplorasi minyak, pipeline perairan dalam, dan banyak area kerja lainnya, terus meningkat kedalaman dan kekompleksitasannya. Seperti ditunjukkan pada gambar 2.13, eksplorasi perairan dalam di teluk Meksiko sudah lebih meningkat selama 2 dekade terakhir, meningkat dari kedalaman 3500 ft (1067 m) pada tahun 1976 menjadi 7600 ft (2316 m) pada tahun 1996. Pengeboran di teluk Meksiko dan lepas Brasil pada kedalaman 3000 ft (914 m) tehitung lebih dari 90 % tersebar di seluruh dunia antara tahun 1985 dan 1997, dan trend ini diharapkan tidak menurun pada waktu mendatang.





Beralih ke perairan dalam untuk eksplorasi gas dan minyak, produksi dan pengembangan telah membuka seluruh pasar baru dalam inofasi pemecahan sistem pada seabed. Eksplorasi telah siap pada kedalaman lebih 10000 ft (3048 m), smentara produksi secara cepat mencapai kedalaman ini. Beralih ke perairan dalam, teluk Meksiko sendiri, telah mengembangkan masalah pengaturan baru yang awalnya hanya ditujukan ke seluruh agensi; keselamatan individu dan lingkungan, karena teknologi berusaha menjaga ketentraman dengan penemuan baru di perairan yang lebih dalam. Perlunya memonitor aktivitas ini di seluruh dunia akan menjaga dari kerusakan kehidupan/lingkungan pantai.

Sebagai tambahan pada industri gas dan minyak lepas pantai, banyaknya tugas yang dikerjakan ROV pada perairan yang lebih dalam. Towed search system, tethered ROV work system, dan autonomous vehicle sekarang secara rutin digunakan pada lokasi dan pencarian objek di seluruh dunia. Kapal seperti CURV III dan ATV dapat mencapai kedalaman 6000 m dan Japan's Kaiko telah mencapai titik kedalaman dilautan pada 10909 m. Dan, dengan menurunkan tingkat biaya, ROV meningkatkan dalam mendukung penelitian ilmiah. Kapal seperti Jason pada Woods Hole Oceanographic Institution dan Tiburon pada Monterey Bay Aquarium Research Institution dapat mendukung penelitian ilmiah sampai kedalaman 6000 m dan 4000m.



Gb. 2.13 Pengeboran di teluk Meksiko



## 2.6 Design ROV

Sistem ROV merupakan kelompok yang saling terkait dengan sub sistem, yang berarti saat berfungsi secara sinergis, akan memperoleh kemampuan yang mengagumkan. Karena karakter dari kapal beraneka ragam, dengan tidak menghilangkan ciri khusus, proses design perlu berulang (iterasi). Karena hubungan saling ketergantungan yang tinggi, kemampuan sistem ROV merupakan sebuah keseimbangan design yang penting dan kesamaan karakteristik operasional. Sistem dapat dibagi kedalam sejumlah sub sistem utama, secara umum termasuk :

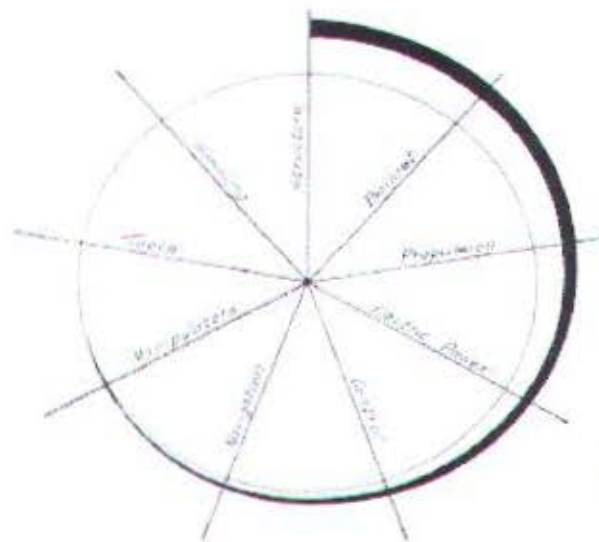
- Kapal, mungkin juga dibagi secara tepat kedalam fungsional sub sistem
- Tools dan sensor
- Kontrol/display console
- Distribusi tenaga listrik
- Kabel umbilical dan tether
- Sistem handling

Hubungan keterkaitan sub sistem secara grafik digambarkan dalam design spiral yang ditunjukkan pada gambar 2.14. Misal, seorang operator mungkin menginginkan meningkatkan kemampuan kecepatan ROV untuk manuver dalam area kondisi geografik tertentu. Penambahan kecepatan yang diinginkan akan berakibat dalam perlunya penambahan energi listrik dan mungkin perlu mengubah thruster/propulsi sub sistem. Penambahan, konfigurasi, struktur, berat, sistem kontrol, ukuran umbilical/tether, sistem manajemen tether winch, sistem handling, dan power supply semuanya akan terpengaruh.





Jelasnya, semua sub sistem dan komponen dipengaruhi dengan apa yang tampaknya merubah peningkatan kemampuan/performance yang diinginkan. Sehingga seorang perancang/operator disarankan menyadari sensitifitas sistem ROV untuk merubah dan berhati-hati mempertimbangkan secara serius perubahan yang mungkin hanya sebagai komponen minor. Keinginan untuk sedikit peningkatan performance/kerusakan dari komponen minor akan sering berdampak pada keseluruhan sistem.



Gb. 2.14 Design spiral ROV

Saat ini, dengan bantuan pengembangan teknologi design komputer, ROV modern telah dikembangkan melalui beberapa iterasi dari design spiral yang ditunjukkan pada gambar 2.14. Saat ini ROV diandalkan dalam melaksanakan pengerjaan kompleks offshore. Bahkan meningkat pada perairan dalam, dan telah mencapai level tinggi dari teknikal design. Kapal ini juga harus fleksibel, oleh karena itu, harus mampu terkonfigurasi untuk beberapa pengoperasian, baik pada sistem kecil maupun besar, yang digunakan untuk beragam inspeksi dan atau pekerjaan. Karena tujuan dari ROV menyelesaikan pekerjaan yang





rumit/komplek, semua kemampuannya biasanya dikemukakan oleh 2 pertimbangan utama yakni persyaratan pekerjaan dan operasional kedalaman air, kedua hal ini mengemukakan pertimbangan dari design spiral.

Bagaimanapun, design dari ROV harus diadakan kedalam seluruh sistem. Beberapa sejumlah besar pertimbangan yang harus dibuat dalam design dan pemilihan sistem ROV seperti :

- Biaya
- Tersedianya teknologi
- Power
- Ukuran
- Berat
- Kedalaman maksimum
- Kondisi laut
- Keamanan
- Keandalan, dll.

#### 2.6.1 Sistem Kontrol

Sistem kontrol digunakan untuk memonitor dan mengontrol ROV ketika ROV dioperasikan dibawah laut ([www.rov.net](http://www.rov.net)). Sistem kontrol berfungsi sebagai berikut :

- ❑ Kontrol propulsi
- ❑ Kontrol kamera bawah laut
- ❑ Kontrol monitor sensor, kedalaman, heading, pitch, dan rolling

Metode tradisional telah menggunakan baik relay (mux) control/computer control, bagaimanapun juga lima tahun terakhir ini orang telah menggunakan PLC



(Programmable Logic Control). Hal ini salah satu faktor otomatisasi yang menjawab pengontrolan motor, masukan dan keluaran sensor eksternal. Pemikiran yang bagus tentang alat ini ialah relatif murah.

Sistem kontrol mengatur kontrol masukan dari operator yang ada pada surface kedalam aksi subsea. Data yang diperlukan oleh operator pada surface untuk keakuratan menentukan posisinya dalam air ditampung oleh sensor dan ditransmisikan kembali ke operator pada surface.

### 2.6.2 Propulsi

Sistem propulsi hadir dalam beberapa tipe yang berbeda ([www.hydrovision.co.uk](http://www.hydrovision.co.uk)), diantaranya :

- Electrical propulsion
- Hydraulic propulsion
- Ducted jet propulsion

Perbedaan tipe ini dikembangkan sesuai dengan ukuran kapal dan tipe operasi yang diharapkan mampu menjalankan operasinya dalam kondisi nyata yang ditentukan dari tipe propulsi yang digunakan.

Tujuan utama dari semua design sistem propulsi adalah thrust yang tinggi ([www.rov.net](http://www.rov.net)).

### 2.7 Pengoperasian Umum ROV ([daveandjill@neteze.com](mailto:daveandjill@neteze.com))

ROV pada dasarnya sebuah miniatur kapal selam yang tak berawak dan memerlukan sistem dasar yang sama yang digunakan pada kapal selam. Supaya ROV menjadi berguna harus dapat manuver saat dibawah laut. Karena lingkungan bawah laut secara esensial merupakan lingkungan 3 dimensi, ROV harus mampu melakukan manuver, maju, mundur, kekiri, kekanan, keatas, dan kebawah.





Kontrol kedalaman dalam menyelam. Diharapkan mempunyai neutral buoyancy. Dengan kata lain, kita menginginkan ROV mempertahankan kedalamannya tanpa tenggelam/muncul kepermukaan. Supaya mendapatkan ini, berat ROV harus sama dengan sejumlah air yang dipindahkan. Lebih jauh lagi, dikehendaki dapat menaikkan/menurunkan ROV. Untuk memperkenankan ROV naik kepermukaan berat ballast harus dibuang dan untuk memperkenankan turun pada kedalaman tertentu berat ballast harus ditambah. Dalam memenuhi ini ROV dibangun dengan tangki ballast yang berupa tangki silinder terempel pada ROV. Ketika tangki ini kosong, berat ROV kurang daripada air yang dipindahkan dan ROV mengapung, supaya ROV tenggelam pada kedalaman yang dikehendaki, air dipompa kedalam tangki. Hal ini membuat berat ROV bertambah, sama dengan sejumlah air yang dipindahkan. Untuk menaikkan ROV kembali, air dipompa keluar dari tangki. Untuk memperkenankan ROV mempertahankan kedalaman tertentu, air secukupnya dipompa kedalam tangki membuat berat ROV sama dengan air yang dipindahkan. Air laut dipompa kedalam dan keluar dari tangki ballast melalui pompa air yang dipasang disamping ROV.

Sewaktu ROV mencapai kedalaman yang diinginkan, ROV tersebut harus digerakkan ke lokasi yang diinginkan. Jika ROV pilot menginginkan ROV bergerak kekiri, dia dapat menggerakkan tongkat kontrol ke kiri. Hal ini akan mengaktifkan water pump dan membuka katup pengoperasian solenoid menggerakkan aliran air keluar melalui nosel kanan yang kemudian mendorong ROV kekiri. Aksi ini sama saat bergerak dalam arah yang lainnya. Ketika ROV diset untuk neutral buoyancy gerakan ROV ke depan, belakang, kiri, dan kanan akan menggerakkan ROV pada bidang horisontal. ROV dapat terbang





keatas/kebawah menggunakan titik keatas dan kebawah yang digabung dengan gerakan ke depan.



### BAB III

## PEMODELAN SISTEM THRUSTER ROV

### 3.1 Tinjauan Umum Sistem

Dalam penulisan tugas akhir ini, yang dibahas adalah penerapan sistem kendali logika fuzzy pada sistem olah gerak ROV terutama pada pengontrolan thruster ROV dalam melakukan manuver saat penjelajahan atau sedang bergerak, dengan menerima masukan baik secara manual dari operator diatas/permukaan ataupun secara otomatis dengan sensor, agar diperoleh keefektifan/kestabilan gerak pada ROV dalam mengatasi perubahan kondisi /bahaya lingkungan dimana ROV tersebut beroperasi.

Pemodelan sistem olah gerak ROV yang dilakukan pada penulisan tugas akhir ini, mengacu pada salah satu produk ROV dari hydrovision yaitu 'Demon' [15].

Adapun karakteristik dari ROV Demon, antara lain : (lebih detailnya terlampir)

- Berat : 3 ton
- Panjang : 2950 mm
- Lebar : 1830 mm
- Tinggi : 2000 mm
- Kecepatan : 4 knot
- Kedalaman kerja : 3000 m

Sedangkan jumlah dan peletakan thruster pada roV, dapat diuraikan sebagai berikut :



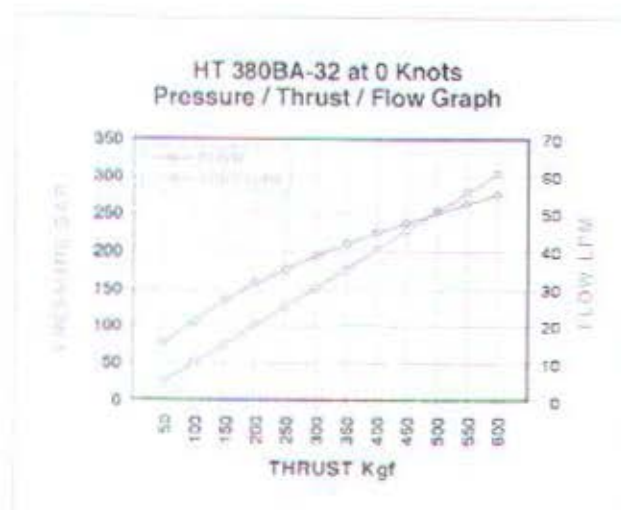
#### A. Jumlah thruster

Thruster yang digunakan pada ROV ini sebanyak 7 buah thruster dengan pembagian 4 buah vectored horizontal thruster dan 3 buah vertical thruster.

Vectored horizontal thruster menggunakan thruster tipe HT 380 BA-32, dengan spesifikasi sebagai berikut :

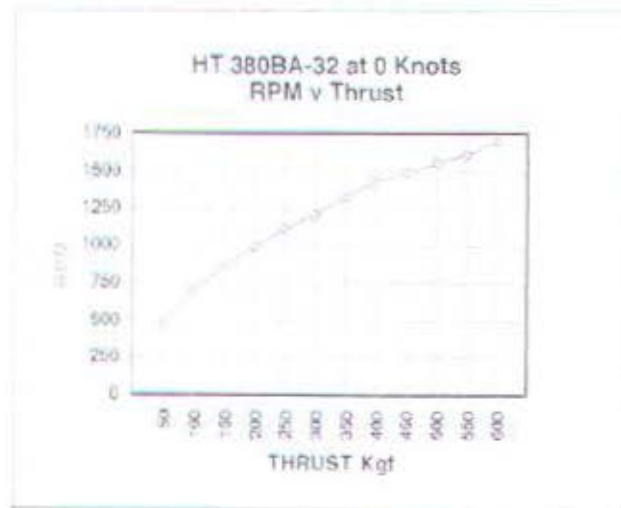
- Static thrust max : 390 Kgf pada 186 bar dan 41 LPM
- Tekanan kerja max : 300 bar
- Diameter propeller : 380 mm
- Berat di udara : 24,5 Kg
- Berat di air : 9,5 Kg
- Kapasitas motor : 32 cc/rev

Sedangkan performance dari thruster tipe HT 380 BA-32 dapat dilihat pada gambar 3.1.

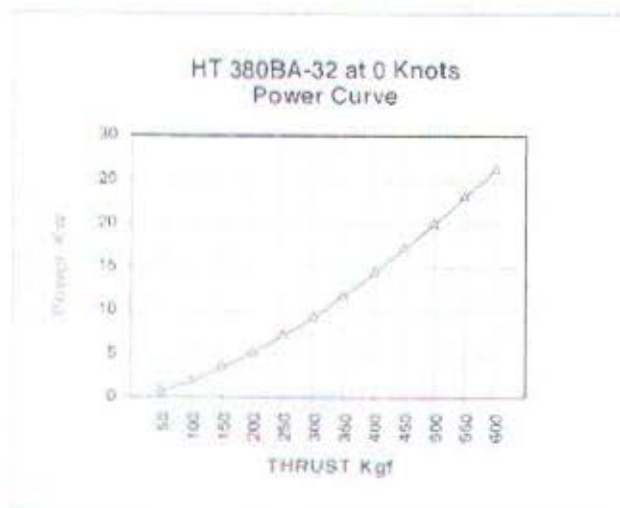


(a)





(b)



(c)

Gb. 3.1 Performance thruster tipe HT 380 BA-32

- (a) Grafik hubungan antara thrust, tekanan, dan aliran
- (b) Grafik hubungan antara thrust dan rpm
- (c) Grafik hubungan antara thrust dan power

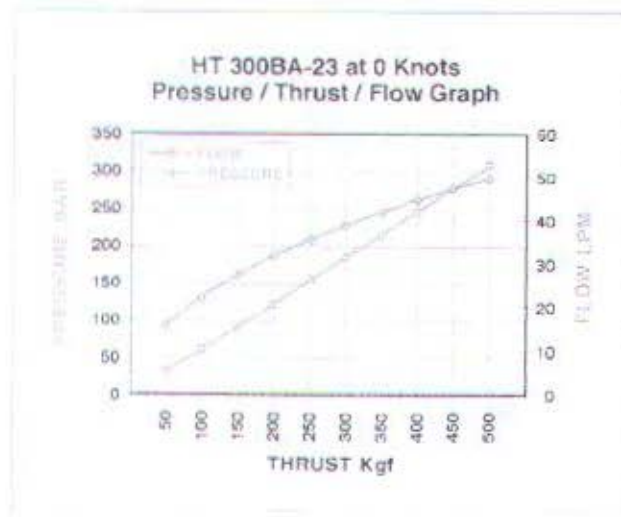
Vertical thruster menggunakan thruster tipe HT 300 BA-32, dengan spesifikasi sebagai berikut :

- Static thrust max : 300 Kgf pada 186 bar dan 39 LPM
- Tekanan kerja max : 300 bar

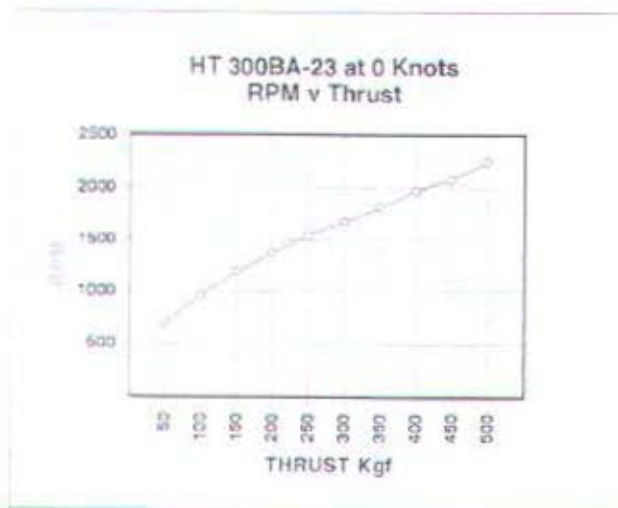


- Diameter propeller : 300 mm
- Berat di udara : 15,5 Kg
- Berat di air : 7 Kg
- Kapasitas motor : 23 cc/rev

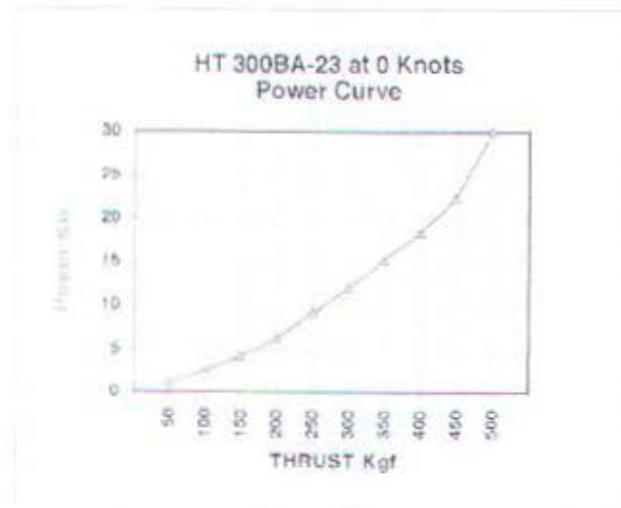
Sedangkan performance dari thruster tipe HT 300 BA-32 dapat dilihat pada gambar 3.2.



(a)



(b)



(c)

Gb. 3.2 Performance thruster tipe HT 300 BA-32

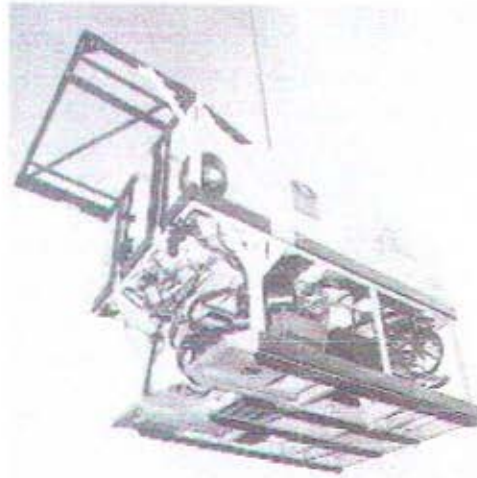
- (a) Grafik hubungan antara thrust, tekanan, dan aliran
- (b) Grafik hubungan antara thrust dan rpm
- (c) Grafik hubungan antara thrust dan power

#### B. Peletakan thruster

Untuk penempatan thruster pada ROV dapat ditunjukkan pada gambar 3.3, dengan pembagian sebagai berikut :

- 4 buah vectored horizontal thruster
  - Aft port thruster
  - Aft stbd thruster
  - Bow port thruster
  - Bow stbd thruster
- 3 buah vertical thruster
  - Aft vertical thruster
  - Bow stbd vertical thruster
  - Bow port vertical thruster





Gb. 3.3 Penempatan thruster ROV

### 3.2 Arsitektur Sistem Kontrol

Sistem kontrol pada ROV menggunakan 2 bagian sistem design, sistem kontrol yang ada pada ROV dan sistem kontrol yang ada pada permukaan, seperti ditunjukkan pada gambar 3.4.

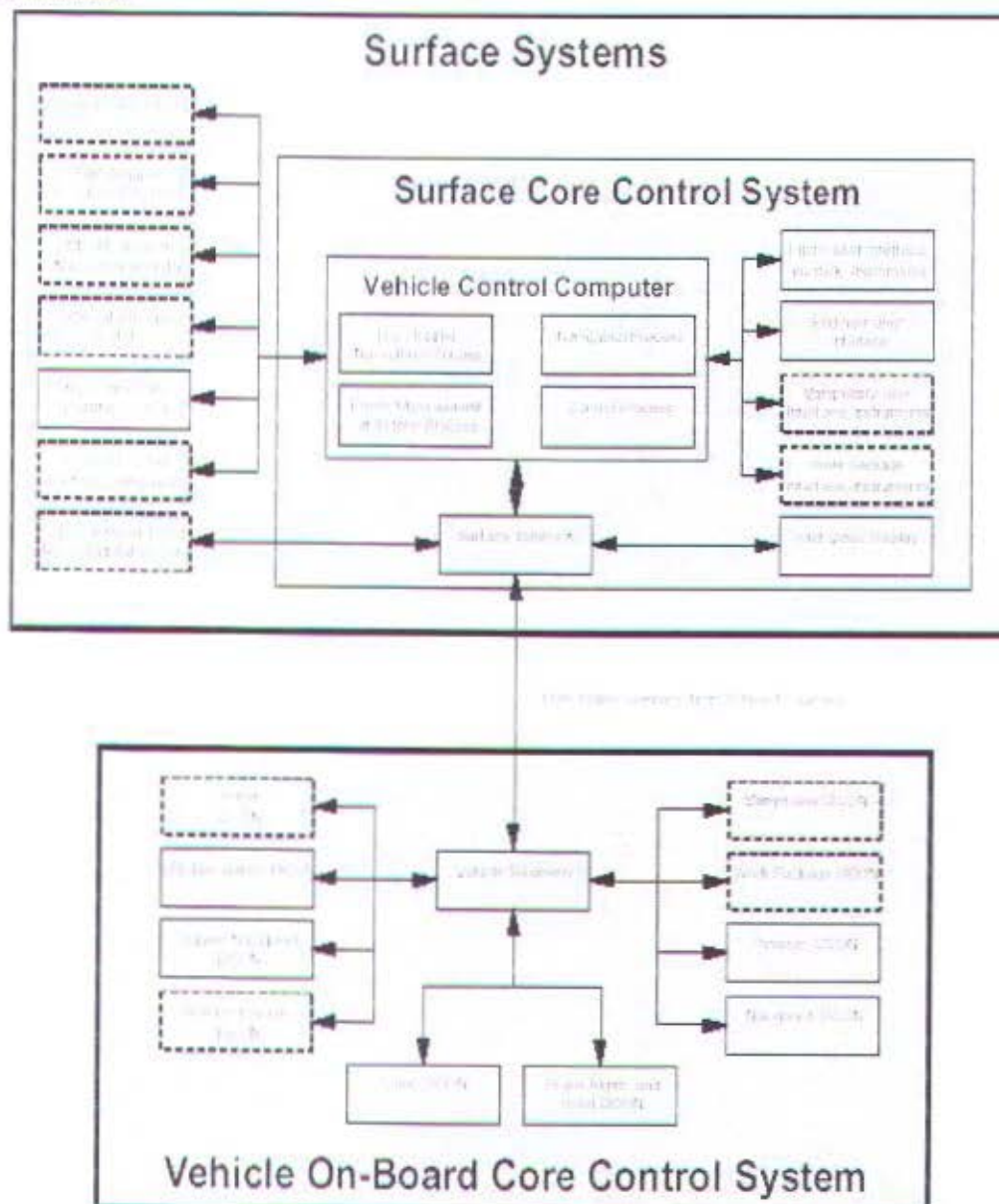
#### ➤ Sistem kontrol pada ROV : Arsitektur pengumpul data (data concentrator)

Sistem kontrol yang ada pada ROV mengontrol dan memonitor semua sensor ROV dan aktuator dalam merespon perintah sesungguhnya (real-time commands) dari sistem kontrol yang ada pada permukaan. Setiap modul pengumpul data akan secara bebas mengontrol power dan data telemetry untuk seluruh subsistem ROV. Pengumpul data akan menerima perintah dari surface control computer, monitor status, dan report data dari instrumen dan subsistem yang ada pada ROV. Setiap pengumpul data bekerja secara asinkron, dan akan menyampaikan ke surface control computer melalui penghubung fiber optik telemetry.



➤ Sistem kontrol pada permukaan

Sistem kontrol yang ada pada permukaan merupakan pusat otak dari sistem kontrol ROV. Sistem kontrol ini menggabungkan semua kontrol yang diperlukan untuk mengoperasikan ROV. Unit kontrol pada permukaan berisi joystick yang memberikan kontrol arah termasuk gerakan maju, mundur, kiri, kanan, atas, dan bawah. Juga mempunyai monitor untuk melihat lingkungan bawah laut.



Gb. 3.4 Arsitektur sistem kontrol ROV

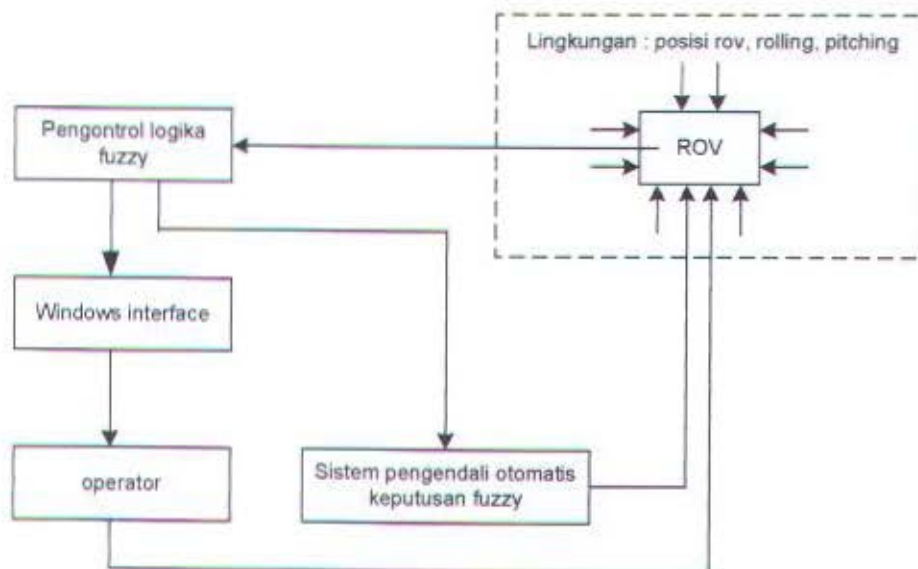


## BAB IV

## PERANCANGAN ALGORITMA KENDALI LOGIKA FUZZY

## 4.1 Arsitektur Pengembangan Sistem Pengendalian

Kerangka dasar pengembangan sistem pengendalian adalah seperti yang ditunjukkan pada gambar 4.1.



Gb. 4.1 Arsitektur sistem pengendalian ROV

Mula-mula ROV menerima masukan berupa informasi keadaan lingkungan sekitar posisi dan keseimbangan ROV itu sendiri, dan juga kecepatan arus yang didefinisikan sebagai akibat dari arus berupa kondisi rolling dan pitching, yang kemudian informasi ini dikuantisasi. Hasil dari kuantisasi dikirim ke pengontrol logika fuzzy, yang kemudian disini diproses sedemikian rupa berdasarkan aturan-aturan yang telah terprogram sehingga muncul sebuah keputusan berupa perbedaan beban thrust yang timbul, baik itu pada horisontal maupun vertikal thruster. Keputusan ini dikirim ke sistem kontrol otomatis katup servo yang memberikan perbedaan beban thrust, baik pada horisontal maupun





vertikal thruster sebesar keputusan pengontrol logika fuzzy. Kesemuanya itu terpantau oleh operator lewat windows interface, sehingga jika terjadi kesalahan keputusan fuzzy, maka operator akan langsung memblokir hubungan antara sistem kontrol otomatis katup servo dengan pengontrol logika fuzzy-nya.

## **4.2 Perancangan Kendali Logika Fuzzy**

### **4.2.1 Penentuan Masukan dan Keluaran Sistem**

Masukan (input) yang diambil dalam penulisan tugas akhir ini, adalah :

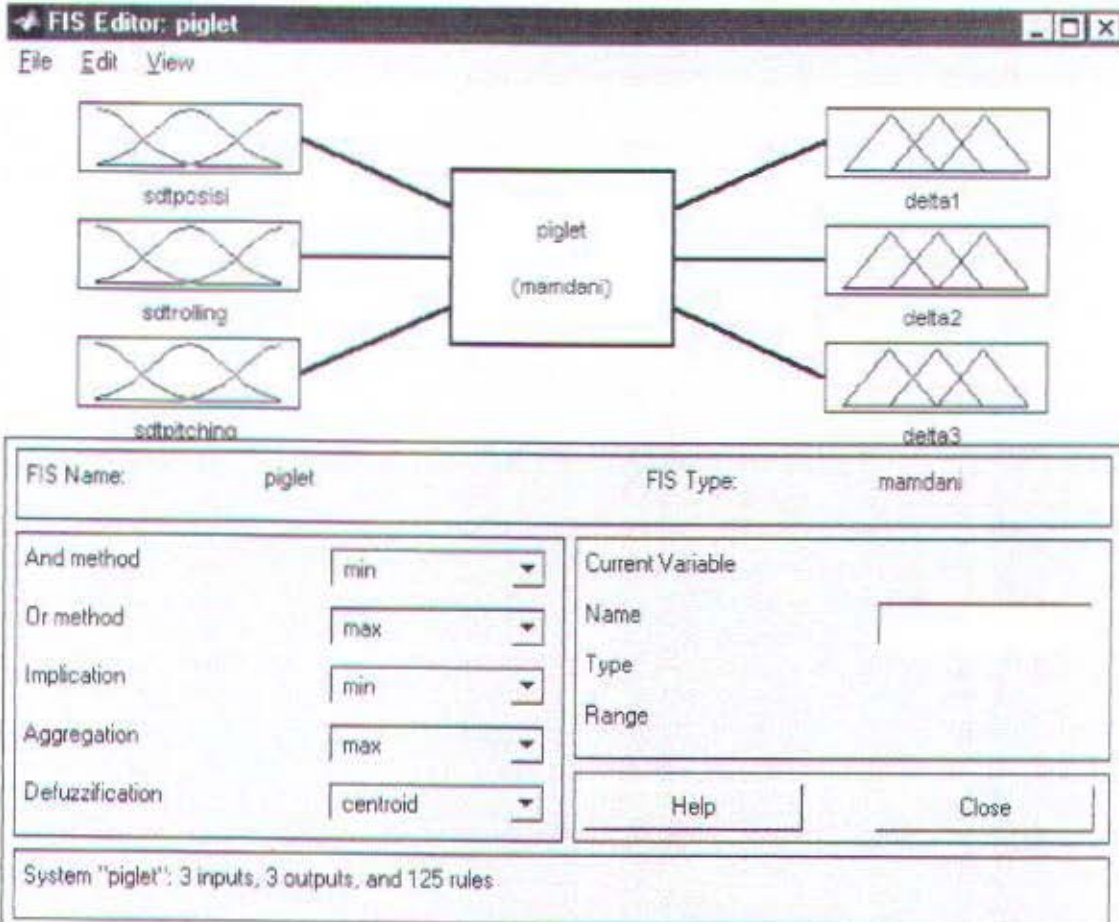
- Sudut posisi ROV
- Sudut rolling ROV
- Sudut pitching ROV

Sedangkan keluaran (output) dari sistem ini, adalah :

- Perbedaan beban thrust antara vectored horizontal thruster port dan stbd
- Perbedaan beban thrust antara vertical thruster port dan stbd
- Perbedaan beban thrust antara vertical thruster depan dan belakang

### **4.2.2 Struktur Sistem Logika Fuzzy**

Berdasarkan variabel-variabel yang ada, maka struktur sistem logika fuzzy yang dikembangkan adalah seperti ditunjukkan pada gambar 4.2.



Gb. 4.2 Struktur sistem logika fuzzy

#### 4.2.3 Penentuan Sifat-Sifat Variabel Linguistik

Sifat-sifat dari masing-masing variabel linguistik adalah seperti yang ditunjukkan pada tabel 4.1 di bawah ini :

Nama Variabel	Sifat-Sifat
Sudut posisi	NB, NK, Z, PK PB
Sudut rolling	NB, NK, Z, PK PB
Sudut pitching	NB, NK, Z, PK PB
delta1	NB, NK, Z, PK PB
delta2	NB, NK, Z, PK PB
delta3	NB, NK, Z, PK PB

Tab. 4.1 Sifat variabel linguistik



Sedangkan detail dari variabel linguistik diatas, seperti yang ditunjukkan pada tabel 4.2 di bawah ini :

Nama variabel	Nilai min	Nilai max	Nilai tengah	Ket
Sudut posisi	-30	30	0	deg
Sudut rolling	-45	45	0	deg
Sudut pitching	-45	45	0	deg
delta1	-390	390	0	kgf
delta2	-300	300	0	kgf
delta3	-300	300	0	kgf

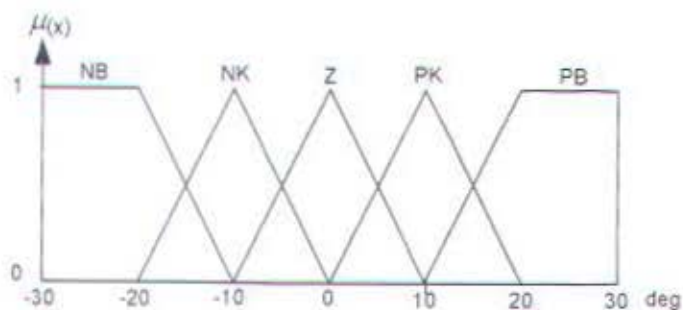
Tab. 4.2 Detail variabel linguistik

#### 4.2.4 Fungsi Keanggotaan

Fungsi keanggotaan yang digunakan dalam penulisan tugas akhir ini adalah bentuk segitga (triangular shape).

##### 4.2.4.1 Fungsi Keanggotaan Variabel Sudut Posisi

Untuk fungsi keanggotaan pada variabel sudut posisi adalah sebagai berikut :



Gb. 4.3 Fungsi keanggotaan sudut posisi

Sedang titik-titik batas keanggotaan masing-masing sifat variabel linguistik sudut posisi adalah sebagai berikut :





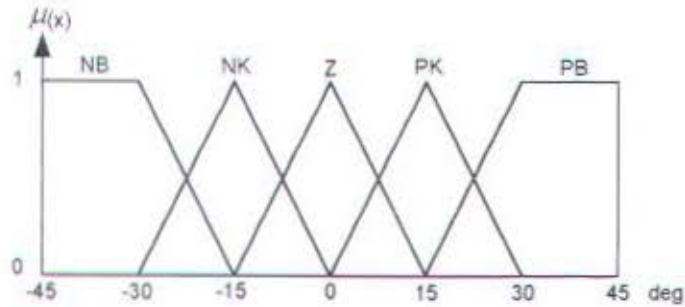
Sifat	Model	Titik batas (x,y)		
NB	linear	(-30,1)	(-20,1)	(-10,0)
		(30,0)		
NK	linear	(-30,0)	(-20,0)	(-10,1)
		(0,0)	(30,0)	
Z	linear	(-30,0)	(-10,0)	(0,1)
		(10,0)	(30,0)	
PK	linear	(-30,0)	(0,0)	(10,1)
		(20,0)	(30,0)	
PB	linear	(-30,0)	(10,0)	(20,1)
		(30,1)		

Tab. 4.3 Titik batas keanggotaan sudut posisi

Dari gambar 4.3 dan tabel 4.3 diatas menunjukkan bahwa sudut posisi ROV  $(-30 \div -20)$  derajat adalah anggota dari 'NB' dengan  $\mu_{(x)} = 1$ , sedangkan  $(-20 \div -10)$  derajat mempunyai 2  $\mu_{(x)}$  yaitu 'NB' dan 'NK' dengan  $(-10)$  derajat adalah anggota 'NK' yang mempunyai  $\mu_{(x)} = 1$ , begitu pula dengan  $(-10 \div 0)$ ,  $(0 \div 10)$ ,  $(10 \div 20)$  derajat masing-masing mempunyai 2  $\mu_{(x)}$ . Untuk  $(20 \div 30)$  derajat adalah anggota 'PB' dengan  $\mu_{(x)} = 1$ .

#### 4.2.4.2 Fungsi Keanggotaan Variabel Sudut Rolling

Untuk fungsi keanggotaan pada variabel sudut rolling adalah sebagai berikut :



Gb. 4.4 Fungsi keanggotaan sudut rolling

Sedang titik-titik batas keanggotaan masing-masing sifat variabel linguistik sudut rolling adalah sebagai berikut :

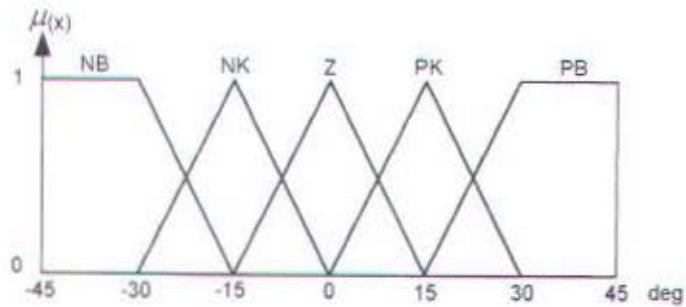
Sifat	Model	Titik batas (x,y)		
NB	linear	(-45,1)	(-30,1)	(-15,0)
		(45,0)		
NK	linear	(-45,0)	(-30,0)	(-15,1)
		(0,0)	(45,0)	
Z	linear	(-45,0)	(-15,0)	(0,1)
		(15,0)	(45,0)	
PK	linear	(-45,0)	(0,0)	(15,1)
		(30,0)	(45,0)	
PB	linear	(-45,0)	(15,0)	(30,1)
		(45,1)		

Tab. 4.4 Titik batas keanggotaan sudut rolling

#### 4.2.4.3 Fungsi Keanggotaan Variabel Sudut Pitching

Untuk fungsi keanggotaan pada variabel sudut pitching adalah sebagai berikut :





Gb. 4.5 Fungsi keanggotaan sudut pitching

Sedang titik-titik batas keanggotaan masing-masing sifat variabel linguistik sudut pitching adalah sebagai berikut :

Sifat	Model	Titik batas (x,y)
NB	linear	(-45,1) (-30,1) (-15,0) (45,0)
NK	linear	(-45,0) (-30,0) (-15,1) (0,0) (45,0)
Z	linear	(-45,0) (-15,0) (0,1) (15,0) (45,0)
PK	linear	(-45,0) (0,0) (15,1) (30,0) (45,0)
PB	linear	(-45,0) (15,0) (30,1) (45,1)

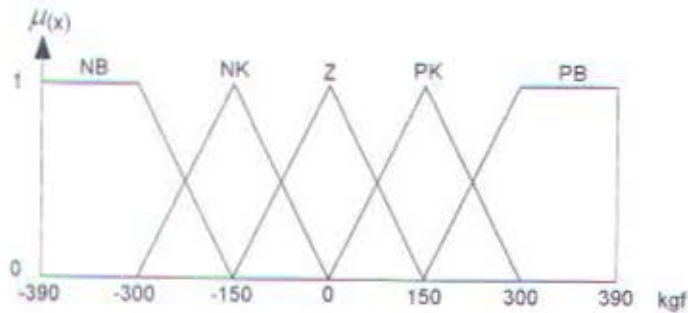
Tab. 4.5 Titik batas keanggotaan sudut pitching





#### 4.2.4.4 Fungsi Keanggotaan Variabel Delta1

Untuk fungsi keanggotaan pada variabel delta1 adalah sebagai berikut :



Gb. 4.6 Fungsi keanggotaan delta1

Sedang titik-titik batas keanggotaan masing-masing sifat variabel linguistik delta1 adalah sebagai berikut :

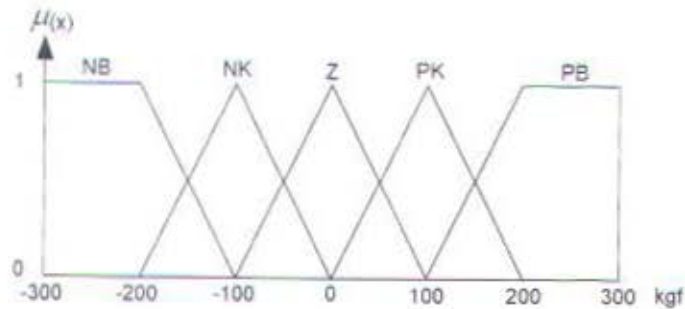
Sifat	Model	Titik batas (x,y)
NB	linear	(-390,1) (-300,1) (-150,0) (390,0)
NK	linear	(-390,0) (-300,0) (-150,1) (0,0) (390,0)
Z	linear	(-390,0) (-150,0) (0,1) (150,0) (390,0)
PK	linear	(-390,0) (0,0) (150,1) (300,0) (390,0)
PB	linear	(-390,0) (150,0) (300,1) (390,1)

Tab. 4.6 Titik batas keanggotaan delta1



#### 4.2.4.5 Fungsi Keanggotaan Variabel Delta2

Untuk fungsi keanggotaan pada variabel delta2 adalah sebagai berikut :



Gb. 4.7 Fungsi keanggotaan delta2

Sedang titik-titik batas keanggotaan masing-masing sifat variabel linguistik delta2 adalah sebagai berikut :

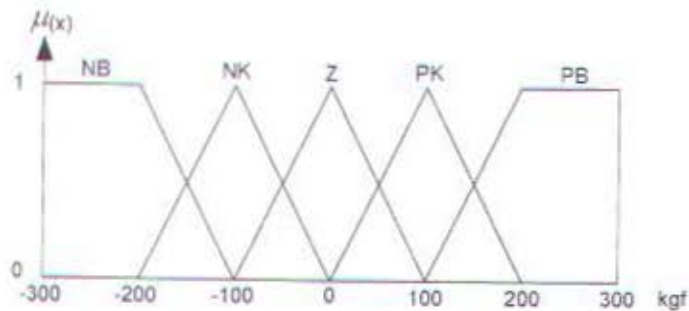
Sifat	Model	Titik batas (x,y)
NB	linear	(-300,1) (-200,1) (-100,0) (300,0)
NK	linear	(-300,0) (-200,0) (-100,1) (0,0) (300,0)
Z	linear	(-300,0) (-100,0) (0,1) (100,0) (300,0)
PK	linear	(-300,0) (0,0) (100,1) (200,0) (300,0)
PB	linear	(-300,0) (100,0) (200,1) (300,1)

Tab. 4.7 Titik batas keanggotaan delta2



#### 4.2.4.6 Fungsi Keanggotaan Variabel Delta3

Untuk fungsi keanggotaan pada variabel delta3 adalah sebagai berikut :



Gb. 4.8 Fungsi keanggotaan delta3

Sedang titik-titik batas keanggotaan masing-masing sifat variabel linguistik delta3 adalah sebagai berikut :

Sifat	Model	Titik batas (x,y)
NB	linear	(-300,1) (-200,1) (-100,0) (300,0)
NK	linear	(-300,0) (-200,0) (-100,1) (0,0) (300,0)
Z	linear	(-300,0) (-100,0) (0,1) (100,0) (300,0)
PK	linear	(-300,0) (0,0) (100,1) (200,0) (300,0)
PB	linear	(-300,0) (100,0) (200,1) (300,1)

Tab.4.8 Titik batas keanggotaan delta3

#### 4.2.5 Pengembangan Rule Base

Pada perancangan sistem pengendalian ini, rule base dibuat dengan menggunakan bantuan software Matlab 5.3.1, sehingga didapatkan rule base seperti pada tabel 4.9.





If				Then		
rule	sudut posisi	sudut rolling	sudut pitching	$\Delta 1$	$\Delta 2$	$\Delta 3$
1	NB	NB	NB	NB	NB	NB
2	NB	NB	NK	NB	NB	NK
3	NB	NB	Z	NB	NB	Z
4	NB	NB	PK	NB	NB	PK
5	NB	NB	PB	NB	NB	PB
6	NB	NK	NB	NB	NK	NB
7	NB	NK	NK	NB	NK	NK
8	NB	NK	Z	NB	NK	Z
9	NB	NK	PK	NB	NK	PK
10	NB	NK	PB	NB	NK	PB
11	NB	Z	NB	NB	Z	NB
12	NB	Z	NK	NB	Z	NK
13	NB	Z	Z	NB	Z	Z
14	NB	Z	PK	NB	Z	PK
15	NB	Z	PB	NB	Z	PB
16	NB	PK	NB	NB	PK	NB
17	NB	PK	NK	NB	PK	NK
18	NB	PK	Z	NB	PK	Z
19	NB	PK	PK	NB	PK	PK
20	NB	PK	PB	NB	PK	PB
21	NB	PB	NB	NB	PB	NB
22	NB	PB	NK	NB	PB	NK
23	NB	PB	Z	NB	PB	Z
24	NB	PB	PK	NB	PB	PK
25	NB	PB	PB	NB	PB	PB



26	NK	NB	NB	NK	NB	NB
27	NK	NB	NK	NK	NB	NK
28	NK	NB	Z	NK	NB	Z
29	NK	NB	PK	NK	NB	PK
30	NK	NB	PB	NK	NB	PB
31	NK	NK	NB	NK	NK	NB
32	NK	NK	NK	NK	NK	NK
33	NK	NK	Z	NK	NK	Z
34	NK	NK	PK	NK	NK	PK
35	NK	NK	PB	NK	NK	PB
36	NK	Z	NB	NK	Z	NB
37	NK	Z	NK	NK	Z	NK
38	NK	Z	Z	NK	Z	Z
39	NK	Z	PK	NK	Z	PK
40	NK	Z	PB	NK	Z	PB
41	NK	PK	NB	NK	PK	NB
42	NK	PK	NK	NK	PK	NK
43	NK	PK	Z	NK	PK	Z
44	NK	PK	PK	NK	PK	PK
45	NK	PK	PB	NK	PK	PB
46	NK	PB	NB	NK	PB	NB
47	NK	PB	NK	NK	PB	NK
48	NK	PB	Z	NK	PB	Z
49	NK	PB	PK	NK	PB	PK
50	NK	PB	PB	NK	PB	PB
51	Z	NB	NB	Z	NB	NB
52	Z	NB	NK	Z	NB	NK
53	Z	NB	Z	Z	NB	Z



54	Z	NB	PK	Z	NB	PK
55	Z	NB	PB	Z	NB	PB
56	Z	NK	NB	Z	NK	NB
57	Z	NK	NK	Z	NK	NK
58	Z	NK	Z	Z	NK	Z
59	Z	NK	PK	Z	NK	PK
60	Z	NK	PB	Z	NK	PB
61	Z	Z	NB	Z	Z	NB
62	Z	Z	NK	Z	Z	NK
63	Z	Z	Z	Z	Z	Z
64	Z	Z	PK	Z	Z	PK
65	Z	Z	PB	Z	Z	PB
66	Z	PK	NB	Z	PK	NB
67	Z	PK	NK	Z	PK	NK
68	Z	PK	Z	Z	PK	Z
69	Z	PK	PK	Z	PK	PK
70	Z	PK	PB	Z	PK	PB
71	Z	PB	NB	Z	PB	NB
72	Z	PB	NK	Z	PB	NK
73	Z	PB	Z	Z	PB	Z
74	Z	PB	PK	Z	PB	PK
75	Z	PB	PB	Z	PB	PB
76	PK	NB	NB	PK	NB	NB
77	PK	NB	NK	PK	NB	NK
78	PK	NB	Z	PK	NB	Z
79	PK	NB	PK	PK	NB	PK
80	PK	NB	PB	PK	NB	PB
81	PK	NK	NB	PK	NK	NB





82	PK	NK	NK	PK	NK	NK
83	PK	NK	Z	PK	NK	Z
84	PK	NK	PK	PK	NK	PK
85	PK	NK	PB	PK	NK	PB
86	PK	Z	NB	PK	Z	NB
87	PK	Z	NK	PK	Z	NK
88	PK	Z	Z	PK	Z	Z
89	PK	Z	PK	PK	Z	PK
90	PK	Z	PB	PK	Z	PB
91	PK	PK	NB	PK	PK	NB
92	PK	PK	NK	PK	PK	NK
93	PK	PK	Z	PK	PK	Z
94	PK	PK	PK	PK	PK	PK
95	PK	PK	PB	PK	PK	PB
96	PK	PB	NB	PK	PB	NB
97	PK	PB	NK	PK	PB	NK
98	PK	PB	Z	PK	PB	Z
99	PK	PB	PK	PK	PB	PK
100	PK	PB	PB	PK	PB	PB
101	PB	NB	NB	PB	NB	NB
102	PB	NB	NK	PB	NB	NK
103	PB	NB	Z	PB	NB	Z
104	PB	NB	PK	PB	NB	PK
105	PB	NB	PB	PB	NB	PB
106	PB	NK	NB	PB	NK	NB
107	PB	NK	NK	PB	NK	NK
108	PB	NK	Z	PB	NK	Z
109	PB	NK	PK	PB	NK	PK



110	PB	NK	PB	PB	NK	PB
111	PB	Z	NB	PB	Z	NB
112	PB	Z	NK	PB	Z	NK
113	PB	Z	Z	PB	Z	Z
114	PB	Z	PK	PB	Z	PK
115	PB	Z	PB	PB	Z	PB
116	PB	PK	NB	PB	PK	NB
117	PB	PK	NK	PB	PK	NK
118	PB	PK	Z	PB	PK	Z
119	PB	PK	PK	PB	PK	PK
120	PB	PK	PB	PB	PK	PB
121	PB	PB	NB	PB	PB	NB
122	PB	PB	NK	PB	PB	NK
123	PB	PB	Z	PB	PB	Z
124	PB	PB	PK	PB	PB	PK
125	PB	PB	PB	PB	PB	PB

Tab. 4.9 Rule base

**Keterangan :**

1. Sudut posisi rov : (N) = arah belok ke kiri  
(P) = arah belok ke kanan
2. Sudut rolling : (N) = searah jarum jam  
(P) = berlawanan arah jarum jam
3. Sudut pitching : (N) = menukik ke belakang  
(P) = menukik ke depan
4.  $\Delta 1$  : perbedaan vectored horizontal thruster antara port dan stbd  
-(P) = port lebih besar daripada stbd  
-(N) = stbd lebih besar daripada port





5.  $\Delta 2$  : perbedaan vertical thruster antara port dan stbd

-(N) = port lebih besar daripada stbd

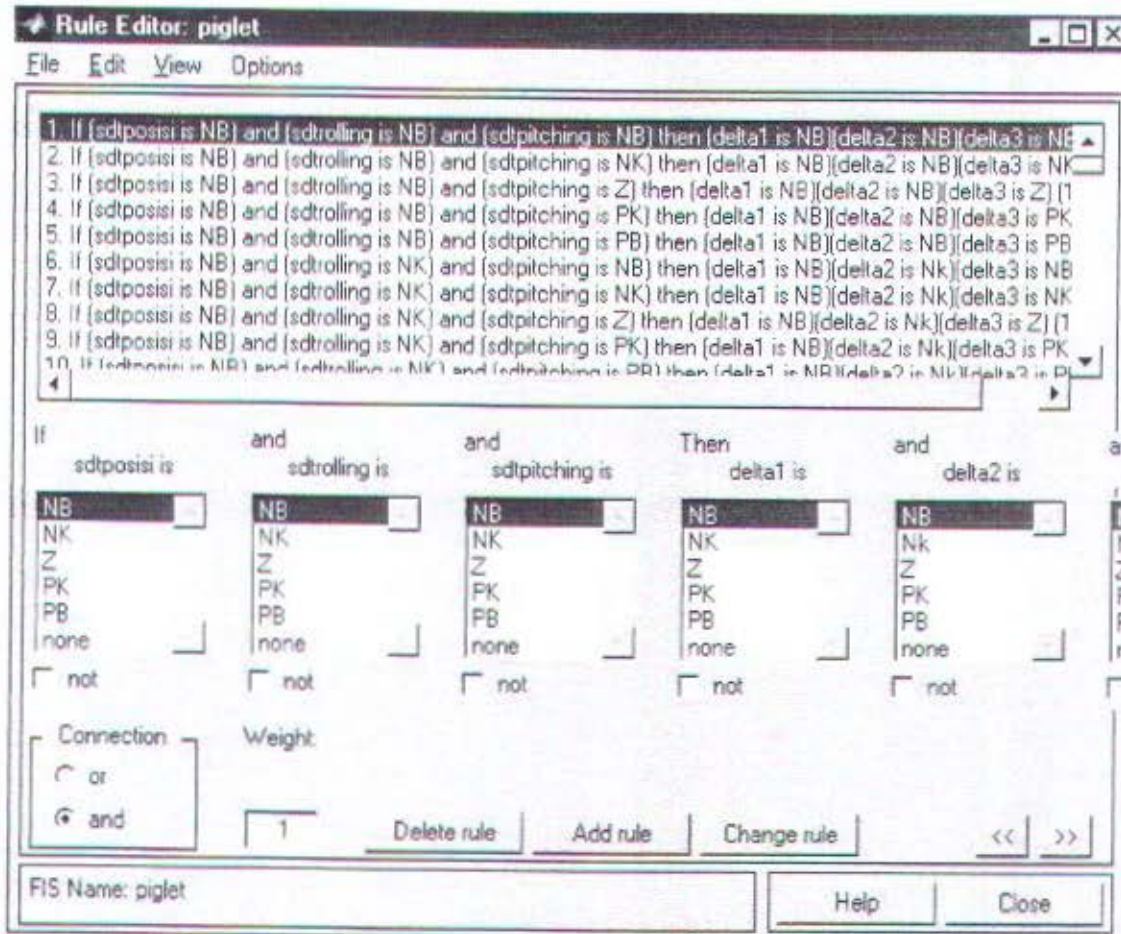
-(P) = stbd lebih besar daripada port

6.  $\Delta 3$  : perbedaan vertical thruster antara depan dan belakang

-(N) = belakang lebih besar daripada depan

-(P) = depan lebih besar daripada belakang

Untuk tampilan rule base pada software Matlab-nya sendiri dapat diperlihatkan pada gambar 4.9 berikut ini :



(a)





(b)

Gb. 4.9 Tampilan rule base pada Matlab

(a) pembuatan rule base

(b) hasil pembuatan rule base

#### 4.3 Sistem Penentuan Thrust dari Masukan

Metode yang digunakan dalam menentukan thrust ROV dari inputan yang masuk adalah dengan menggunakan metode defuzzifikasi COA (Centre of Area). Cara penentuan thrust dengan metode ini adalah sebagai berikut :

Misal, seorang operator menginginkan ROV berbelok ke kanan sebesar 10 derajat, dan pada saat itu terjadi rolling dan pitching pada ROV tersebut sebesar masing-masing 5 derajat yang ditangkap oleh sensor untuk rolling dan pitching.

Langkah-langkah yang harus dilakukan adalah sebagai berikut :



a) Tinjauan terhadap masing-masing variabel linguistik

1. Tinjauan terhadap sudut posisi

Sudut posisi dengan besar 10 derajat, berdasarkan gambar 4.3 dan tabel 4.3 merupakan anggota dari 'PK' dengan derajat keanggotaan  $(\mu_{(x)}) = 1$ .

2. Tinjauan terhadap sudut rolling

Sudut rolling dengan besar 5 derajat, berdasarkan gambar 4.4 dan tabel 4.4 merupakan anggota dari 'PK' dengan derajat keanggotaan  $(\mu_{(x)})$  :

- Untuk 'Z' :  $\frac{15-5}{15-0} = 0,67$  dan

- Untuk 'PK' :  $\frac{5-0}{15-0} = 0,33$

3. Tinjauan terhadap sudut pitching

Sudut pitching dengan besar 5 derajat, berdasarkan gambar 4.5 dan tabel 4.5 merupakan anggota dari 'Z' dan 'PK' dengan derajat keanggotaan  $(\mu_{(x)})$  :

- Untuk 'Z' :  $\frac{15-5}{15-0} = 0,67$  dan

- Untuk 'PK' :  $\frac{5-0}{15-0} = 0,33$

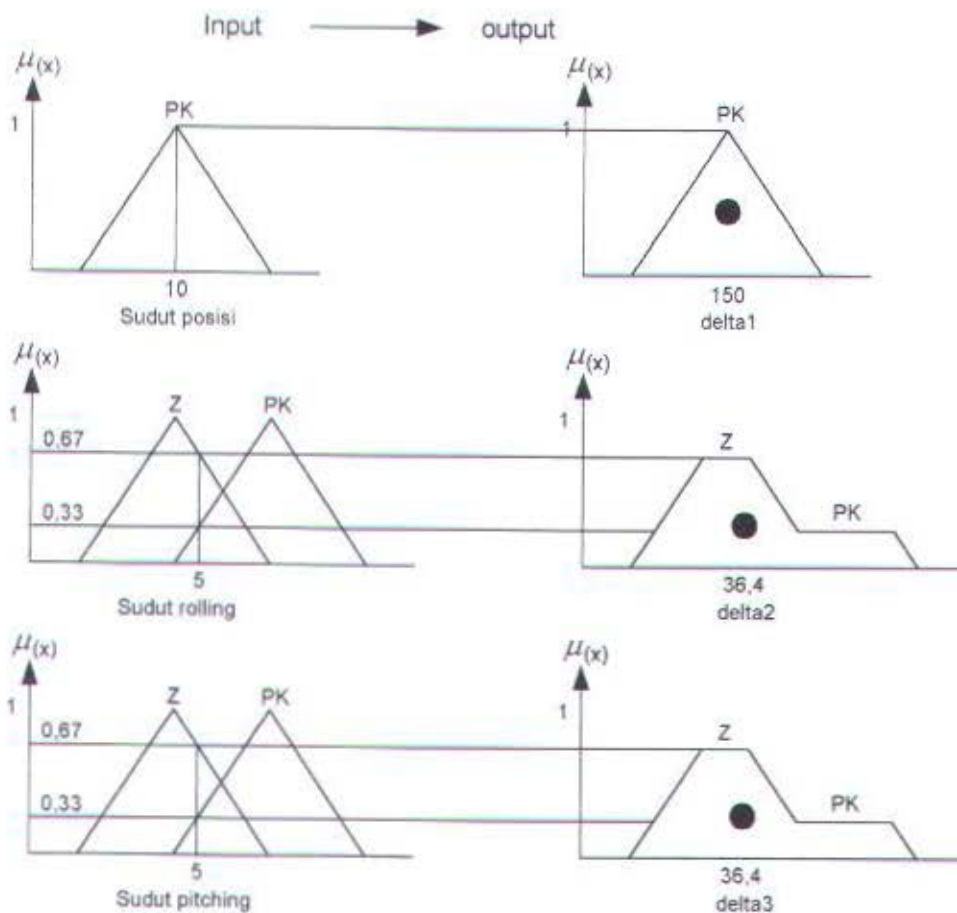
b) Berdasarkan keterangan diatas, maka dapat ditarik kesimpulan bahwa rule base yang digunakan adalah sebagai berikut :

- If sudut posisi 'PK' and sudut rolling 'Z' and sudut pitching 'Z' then delta1 'PK' and delta2 'Z' and delta3 'Z'.
- If sudut posisi 'PK' and sudut rolling 'Z' and sudut pitching 'PK' then delta1 'PK' and delta2 'Z' and delta3 'PK'.



- If sudut posisi 'PK' and sudut rolling 'PK' and sudut pitching 'Z' then delta1 'PK' and delta2 'PK' and delta3 'Z'.
- If sudut posisi 'PK' and sudut rolling 'PK' and sudut pitching 'PK' then delta1 'PK' and delta2 'PK' and delta3 'PK'.

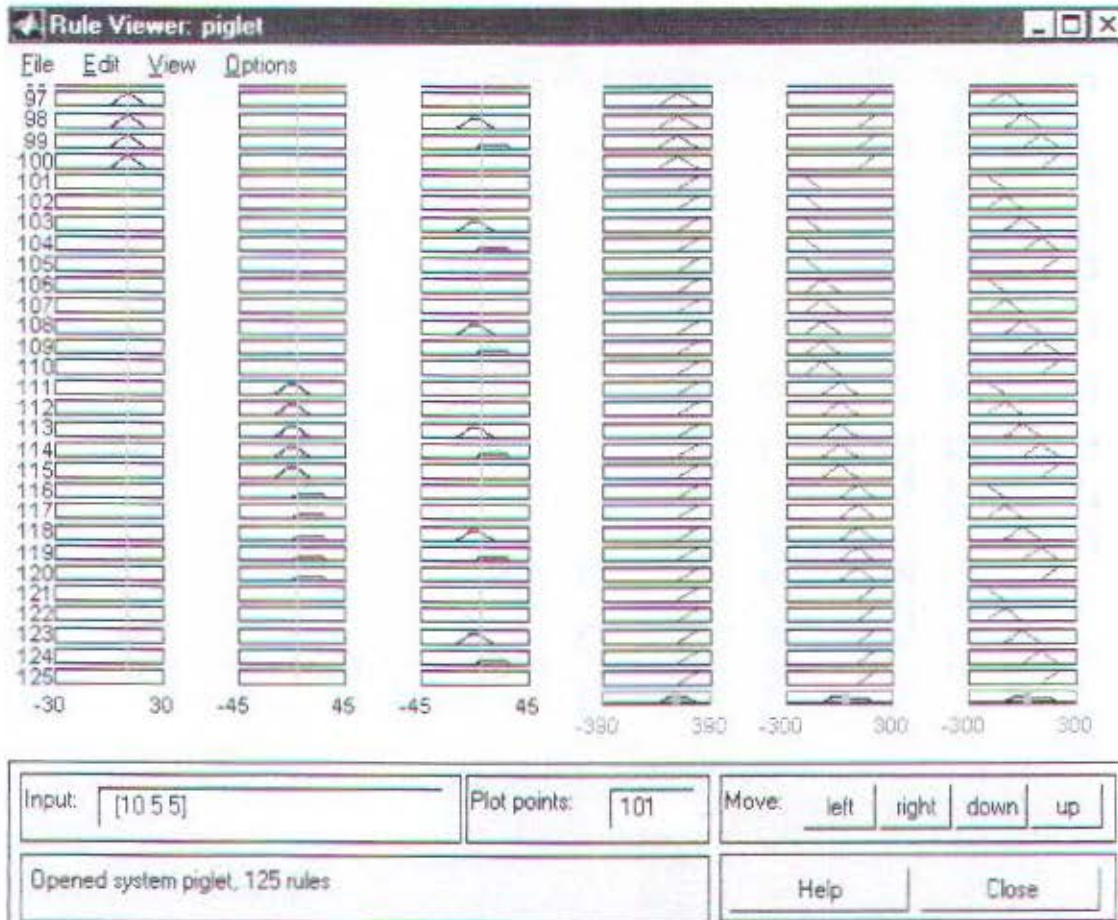
Dengan bantuan software Matlab, maka didapatkan besarnya thrust yang diinginkan untuk mengatasi kondisi yang terjadi seperti ditunjukkan pada gambar 4.10.



Gb. 4.10 Metode penentuan thrust

Untuk tampilan metode penentuan thrust ini dapat diperlihatkan pada gambar 4.9(b) dan gambar 4.11.





Gb. 4.11 Tampilan metode penentuan thrust pada Matlab



## BAB V

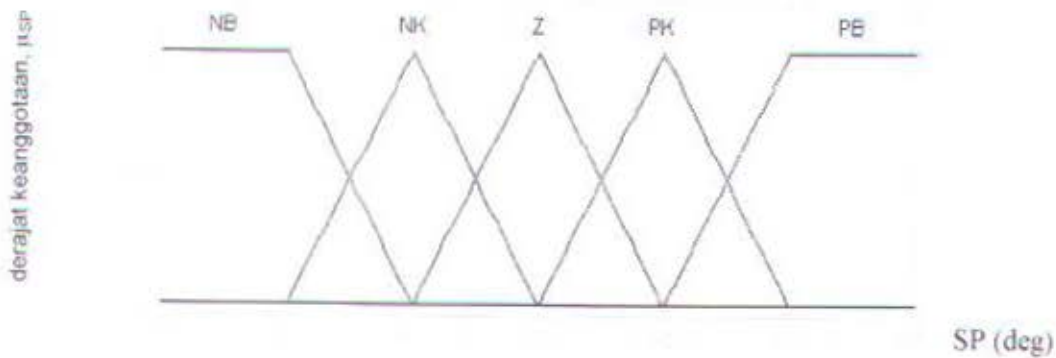
### ANALISA HASIL PERANCANGAN

Setelah dilakukan perancangan algoritma controller logika fuzzy dengan menggunakan bantuan dari pemograman Matlab 5.3.1, maka selanjutnya dilakukan simulasi (hanya berupa tampilan dalam bentuk tabel dan grafik) dan kemudian dilakukan analisa hasil perancangan berdasarkan hasil simulasi yang dilakukan.

#### 5.1 Simulasi

Simulasi yang dilakukan dalam penulisan tugas akhir ini hanya sebatas pada tampilan perhitungan yang disajikan dalam bentuk tabel dan grafik, seperti yang diperlihatkan pada tabel dan grafik 5.1 sampai dengan 5.3.

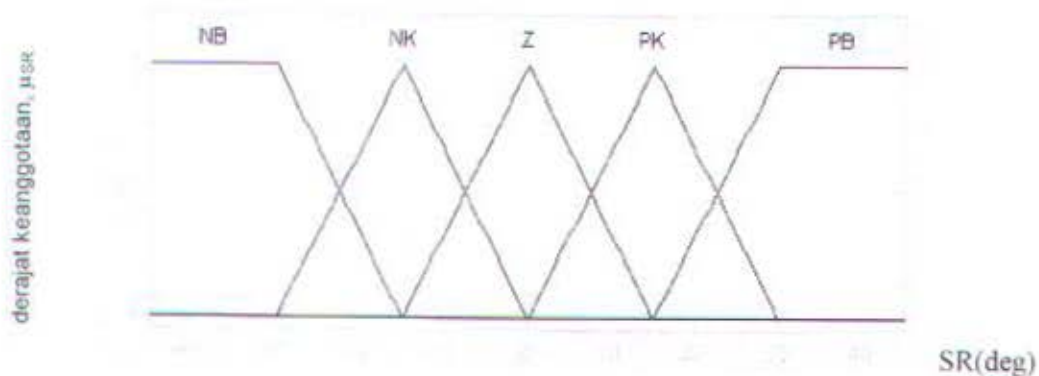
Setelah dilakukan penentuan variabel masukan perancangan logika fuzzy yaitu sudut posisi (SP), sudut rolling (SR), dan sudut pitching (SPt), serta variabel keluaran yaitu perbedaan thrust antar vectored horizontal thruster port dan stbd ( $\Delta 1$ ), perbedaan thrust antara vertical thruster port dan stbd ( $\Delta 2$ ), dan perbedaan thrust antara vertical thruter depan dan belakang ( $\Delta 3$ ), dengan fungsi keanggotaan segitiga seperti yang telah disebutkan pada bab IV, selanjutnya hal tersebut diimplementasikan ke dalam program Matlab seperti yang ditampilkan pada gambar 5.1 sampai dengan 5.6.



Gb. 5.1 Tampilan program Matlab – Fungsi keanggotaan sudut posisi

Dalam gambar 5.1 diatas merupakan implementasi dari fungsi keanggotaan variabel masukan berupa sudut posisi pada program Matlab, disini ditentukan :

- 5 variabel linguistik, yaitu NB, NK, Z, PK, PB.
- Range segmen untuk variabel masukan berupa sudut posisi ditentukan sebesar  $[-30 \ 30]$ . Nilai ini merupakan nilai yang ditentukan oleh perancang menurut pengetahuan dan pengalaman si perancang berdasarkan karakteristik dari obyek yang diatur.



Gb. 5.2 Tampilan program Matlab – Fungsi keanggotaan sudut rolling

Dalam gambar 5.2 diatas merupakan implementasi dari fungsi keanggotaan variabel masukan berupa sudut rolling pada program Matlab, disini ditentukan :

- 5 variabel linguistik, yaitu NB, NK, Z, PK, PB.

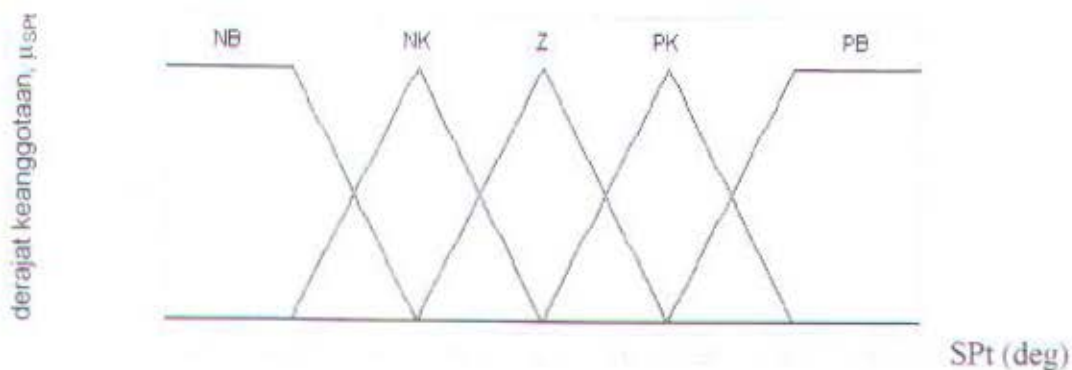




- Range segmen untuk variabel masukan berupa sudut rolling ditentukan sebesar  $[-45 \ 45]$ . Nilai ini merupakan nilai yang ditentukan oleh perancang menurut pengetahuan dan pengalaman si perancang berdasarkan karakteristik dari obyek yang diatur.

Dalam gambar 5.3 berikut ini merupakan implementasi dari fungsi keanggotaan variabel masukan berupa sudut pitching pada program Matlab, disini ditentukan :

- 5 variabel linguistik, yaitu NB, NK, Z, PK, PB.
- Range segmen untuk variabel masukan berupa sudut pitching ditentukan sebesar  $[-45 \ 45]$ . Nilai ini merupakan nilai yang ditentukan oleh perancang menurut pengetahuan dan pengalaman si perancang berdasarkan karakteristik dari obyek yang diatur.



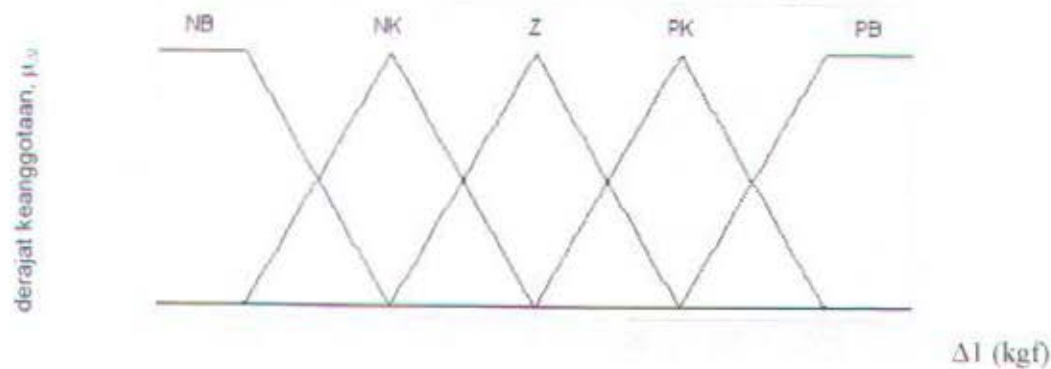
Gb. 5.3 Tampilan program Matlab – Fungsi keanggotaan sudut pitching

Dalam gambar 5.4 berikut ini merupakan implementasi dari fungsi keanggotaan variabel keluaran berupa perbedaan thrust antar vectored horizontal thruster port dan stbd ( $\Delta 1$ ) pada program Matlab, disini ditentukan :

- 5 variabel linguistik, yaitu NB, NK, Z, PK, PB.
- Range segmen untuk variabel keluaran berupa perbedaan thrust antar vectored horizontal thruster port dan stbd ( $\Delta 1$ ) ditentukan sebesar  $[-390 \ 390]$ . Nilai ini



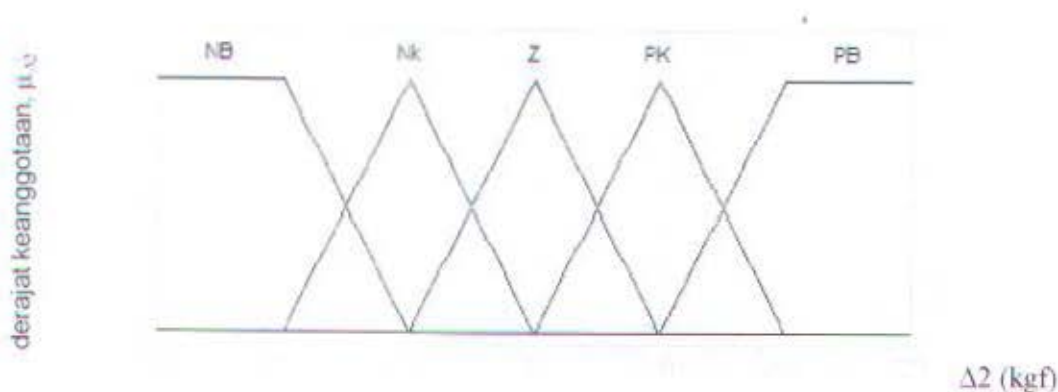
merupakan nilai yang ditentukan oleh perancang menurut pengetahuan dan pengalaman si perancang berdasarkan karakteristik dari obyek yang diatur.



Gb. 5.4 Tampilan program Matlab – Fungsi keanggotaan  $\Delta 1$

Dalam gambar 5.5 berikut ini merupakan implementasi dari fungsi keanggotaan variabel keluaran berupa perbedaan thrust antara vertical thruster port dan stbd ( $\Delta 2$ ) pada program Matlab, disini ditentukan :

- 5 variabel linguistik, yaitu NB, NK, Z, PK, PB.
- Range segmen untuk variabel keluaran berupa perbedaan thrust antara vertical thruster port dan stbd ( $\Delta 2$ ) ditentukan sebesar  $[-300 \ 300]$ . Nilai ini merupakan nilai yang ditentukan oleh perancang menurut pengetahuan dan pengalaman si perancang berdasarkan karakteristik dari obyek yang diatur.

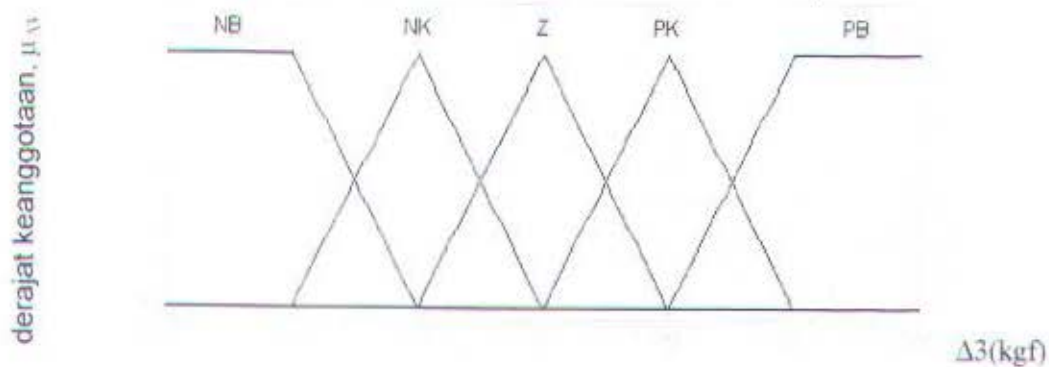


Gb. 5.5 Tampilan program Matlab – Fungsi keanggotaan  $\Delta 2$



Dalam gambar 5.6 berikut ini merupakan implementasi dari fungsi keanggotaan variabel keluaran berupa perbedaan thrust antara vertical thruster depan dan belakang ( $\Delta 3$ ) pada program Matlab, disini ditentukan :

- 5 variabel linguistik, yaitu NB, NK, Z, PK, PB.
- Range segmen untuk variabel keluaran berupa perbedaan thrust antara vertical thruster depan dan belakang ( $\Delta 3$ ) ditentukan sebesar  $[-300 \ 300]$ . Nilai ini merupakan nilai yang ditentukan oleh perancang menurut pengetahuan dan pengalaman si perancang berdasarkan karakteristik dari obyek yang diatur.



Gb. 5.6 Tampilan program Matlab – Fungsi keanggotaan  $\Delta 3$

Langkah selanjutnya setelah implementasi fungsi keanggotaan setiap variabel masukan dan keluaran yang telah kita tentukan terlebih dulu kedalam program Matlab, kemudian kita mulai menyusun aturan fuzzy (rule base fuzzy) yang diperlukan untuk pengontrolan obyek yang kita ingin atur berdasarkan pengetahuan dan pengalaman dari si perancang menurut karakteristik dari obyek yang kita atur dengan menggunakan variabel linguistik, seperti yang telah diuraikan pada bab IV. Setelah dilakukan pengembangan aturan logika fuzzy, maka selanjutnya kita lakukan proses defuzzifikasi untuk mendapatkan hasil/kondisi sebenarnya dari obyek yang kita atur (dalam penulisan tugas akhir ini, kita akan mendapatkan berapa besarnya thrust yang diperlukan berdasarkan





besarnya masukan yang diterima). Hasil yang kita dapat setelah melalui proses defuzzifikasi ini, kemudian kita simulasikan untuk mengetahui berapa lama waktu yang diperlukan oleh obyek yang kita atur untuk dapat mencapai kondisi yang kita inginkan berdasarkan masukan yang diterimanya.

### 5.1.1 Simulasi Terhadap Sudut Posisi (SP)

Dalam melakukan simulasi terhadap sudut posisi ini, dilakukan pendekatan dengan menggunakan persamaan umum gerak (*Rawson, 1984*), seperti berikut ini :

$$\dot{SP} + 2k_{\pm}\omega_0 SP + \omega_0^2 SP = \Delta I \times r_{sp} \dots\dots\dots(5.1a)$$

dimana :

$$\begin{aligned} I_z &= \frac{D}{g} \left( \frac{B^2 \alpha^2}{11.4\delta} + \frac{H^2}{12} \right) \\ &= \frac{3000}{9.8} \left[ \left( \frac{(1.830)^2 \times (1)^2}{11.4 \times 1} \right) + \frac{(2)^2}{12} \right] \\ &= 191.97 \text{ kg.dt}^2.\text{m} \end{aligned}$$

$$\begin{aligned} I_z &= \frac{D}{g} (k_{\pm})^2 \\ k_{\pm} &= \sqrt{\frac{I_z \times g}{D}} = \sqrt{\frac{191.97 \times 9.8}{3000}} \\ &= 0.792 \text{ m} \end{aligned}$$

$$\begin{aligned} \omega_0^2 &= \frac{c}{I_z} = \frac{\rho g L B \alpha \times g}{\rho g L B H \delta (k_{\pm})^2} \\ &= \frac{g}{H (k_{\pm})^2} = \frac{9.8}{2(0.792)^2} \\ &= 7.81 \text{ (rad/dt)}^2 \end{aligned}$$



$$\omega_o = 2.79 \text{ rad/dt}$$

sehingga persamaan (5.1a) dapat dituliskan sebagai berikut :

$$\dot{SP} + 4.42 SP + 7.81 SP = \Delta I \times r_{SP} \dots\dots\dots (5.1b)$$

dengan mengubah persamaan (5.1b) kedalam fungsi bilangan kompleks, maka didapatkan :

$$\frac{SP(s)}{\Delta I \times r_{SP}} = \frac{1}{s^2 + 4.42s + 7.81} \dots\dots\dots (5.1c)$$

$$s_{1,2} = \frac{-4.42 \pm \sqrt{19.54 - 31.24}}{2}$$

$$= \frac{-4.42 \pm \sqrt{-11.7}}{2} = -2.21 \pm 1.71j$$

$$\frac{a1(s - \alpha) + a2\beta}{(s - \alpha)^2 + \beta^2} = \frac{a1(s + 2.21) + 1.71a2}{(s + 2.21)^2 + (1.71)^2}$$

$$a1(s + 2.21) + 1.71a2 = 1 \Big|_{s=-2.21+1.71j}$$

$$a1(-2.21 + 1.71j + 2.21) + 1.71a2 = 1$$

$$a1 = 0 \quad / \quad a2 = \frac{1}{1.71}$$

persamaan (5.1c), dapat dituiskan kembali menjadi :

$$\frac{SP(s)}{\Delta I \times r_{SP}} = \frac{1}{(s + 2.21)^2 + (1.71)^2} \dots\dots\dots (5.1d)$$

kemudian persamaan (5.1d) diubah kedalam fungsi waktu yang kemudian akan disimulasikan, sebagai berikut :

$$SP(t) = \frac{\Delta I \times r_{SP} \times e^{-2.21t} \times \sin 1.71t}{1.71} \dots\dots\dots (5.1e)$$

dimana : -  $\Delta I$  merupakan hasil dari defuzzifikasi berdasarkan masukan dari SP

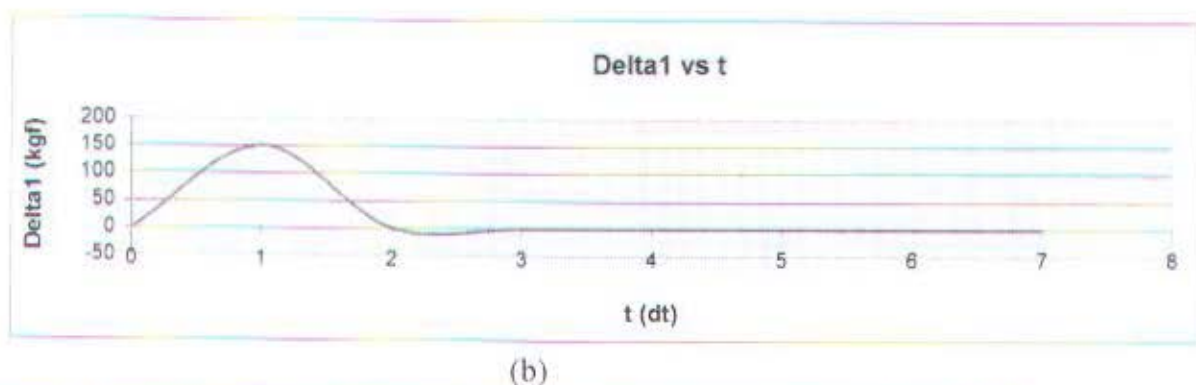
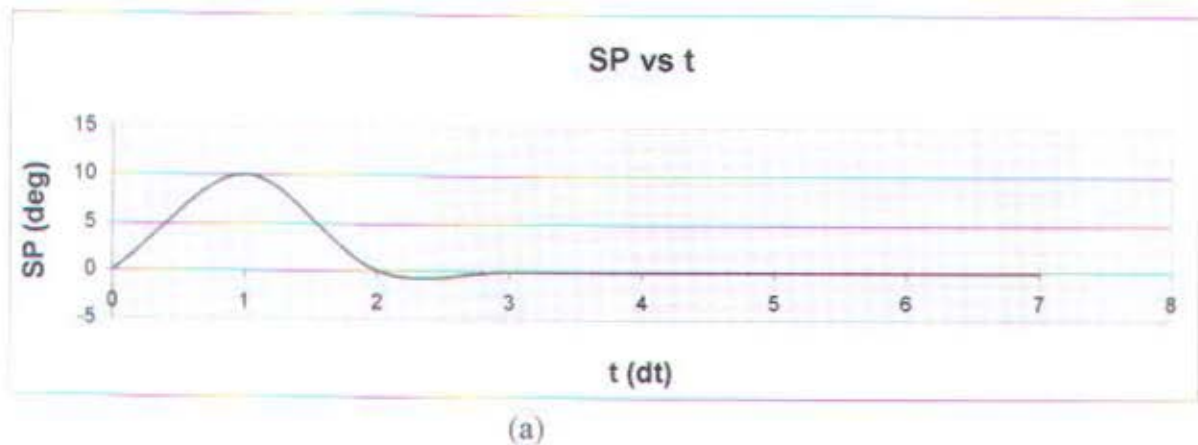


$$-r_{SP} = 0.725 \text{ m}$$

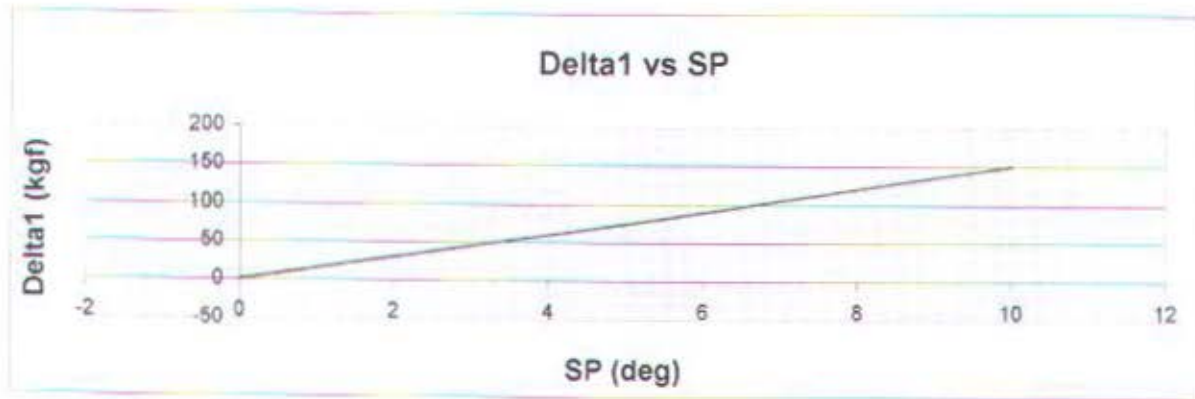
Dengan menggunakan persamaan (5.1e), maka simulasi terhadap sudut posisi didapatkan seperti yang diperlihatkan pada tabel dan grafik 5.1 berikut ini :

SP (deg)	t (dt)	$\Delta 1$ (kgf)	$r_{SP}$ (m)	$e^{-2.21t}$	$\sin 1.71t$
0	0	0	-	-	-
10	1	150	-	-	-
0.045656	2	1.02	0.725	0.012034	0.059655
5.1E-05	3	0.00114	0.725	0.00132	0.089416
8.34E-09	4	1.86E-07	0.725	0.000145	0.119097
1.86E-13	5	4.16E-12	0.725	1.59E-05	0.148672
5.48E-19	6	3.30E-15	0.725	1.74E-06	0.178115
5.55E-23	7	3.30E-15	0.725	1.91E-07	0.2074

Tab. 5.1 Tabel simulasi terhadap sudut posisi







(c)

Graf. 5.1 Grafik simulasi terhadap sudut posisi

(a) Hubungan antara SP vs t

(b) Hubungan antara delta l vs t

(c) Hubungan antara delta l vs SP

### 5.1.2 Simulasi Terhadap Sudut Rolling (SR)

Dalam melakukan simulasi terhadap sudut rolling ini, dilakukan pendekatan dengan menggunakan persamaan umum gerak (*Rawson, 1984*), seperti berikut ini :

$$\ddot{SR} + 2k_{xx}\omega_0 SR + \omega_0^2 SR = \Delta z \times r_{SR} \dots\dots\dots(5.2a)$$

dimana :

- $k_{xx} \rightarrow 0.33B \leq k_{xx} \leq 0.45B$  (*Rameswar, 1978*)

$$k_{xx} = 0.40B = 0.40 \times 1.830$$

$$= 0.732 \text{ m}$$

- $I_x = \frac{D}{g}(k_{xx})^2 = \frac{3000}{9.8}(0.732)^2$

$$= 164.03 \text{ kg} \cdot \text{dt}^2 \cdot \text{m}$$

- $(r-a) \cong r = \frac{\alpha^2}{14\delta} \times \frac{B^2}{H}$



$$= \frac{1^2}{14 \times 1} \times \frac{(1.830)^2}{2}$$

$$= 0.12 \text{ m}$$

$$\omega_n^2 = \frac{D(r-a)}{I_x}$$

$$= \frac{3000 \times 0.12}{164.03} = 2.19 \text{ (rad/dt)}^2$$

$$\omega_n = 1.48 \text{ rad/dt}$$

sehingga persamaan (5.2a) dapat dituliskan sebagai berikut :

$$\dot{SR} + 2.17SR + 2.19SR = \Delta 2 \times r_{SR} \dots\dots\dots (5.2b)$$

dengan mengubah persamaan (5.2b) kedalam fungsi bilangan kompleks, maka didapatkan :

$$\frac{SR(s)}{\Delta 2 \times r_{SR}} = \frac{1}{s^2 + 2.17s + 2.19} \dots\dots\dots (5.2c)$$

$$s_{1,2} = \frac{-2.17 \pm \sqrt{4.71 - 8.76}}{2}$$

$$= \frac{-2.17 \pm \sqrt{-4.05}}{2} = -1.09 \pm 1.01j$$

$$\frac{a1(s-\alpha) + a2\beta}{(s-\alpha)^2 + \beta^2} = \frac{a1(s+1.09) + 1.01a2}{(s+1.09)^2 + (1.01)^2}$$

$$a1(s+1.09) + 1.01a2 = 1 \Big|_{s=-1.09+1.01j}$$

$$a1(-1.09+1.01j+1.09) + 1.01a2 = 1$$

$$a1 = 0 \quad / \quad a2 = \frac{1}{1.01}$$

persamaan (5.2c), dapat dituiskan kembali menjadi :





$$\frac{SR(s)}{\Delta 2 \times r_{SR}} = \frac{1}{(s+1.09)^2 + (1.01)^2} \dots\dots\dots (5.2d)$$

kemudian persamaan (5.2d) diubah kedalam fungsi waktu yang kemudian akan disimulasikan, sebagai berikut :

$$SR(t) = \frac{\Delta 2 \times r_{SR} \times e^{-1.09t} \times \sin 1.01t}{1.01} \dots\dots\dots (5.2e)$$

dimana : -  $\Delta 2$  merupakan hasil dari defuzzifikasi berdasarkan masukan dari SR

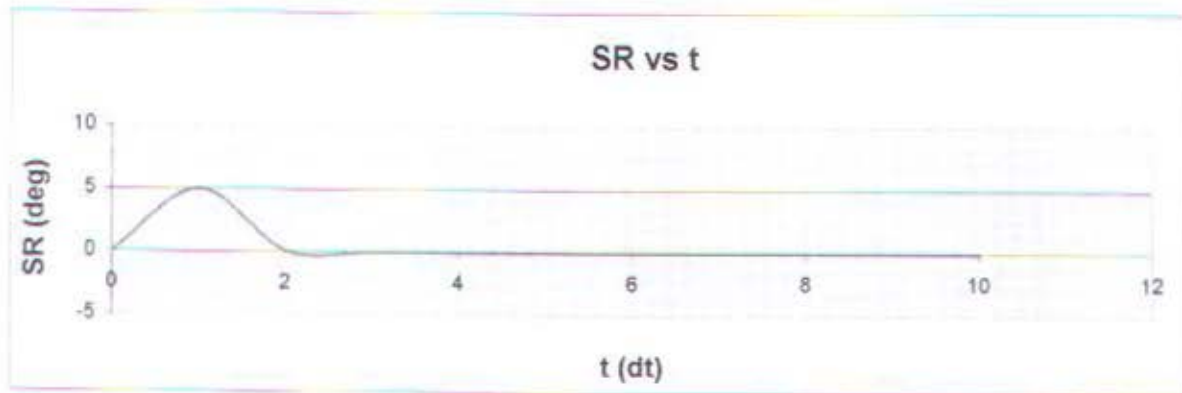
$$- r_{SR} = 0.765 \text{ m}$$

Dengan menggunakan persamaan (5.2e), maka simulasi terhadap sudut rolling didapatkan seperti yang diperlihatkan pada tabel dan grafik 5.2 berikut ini :

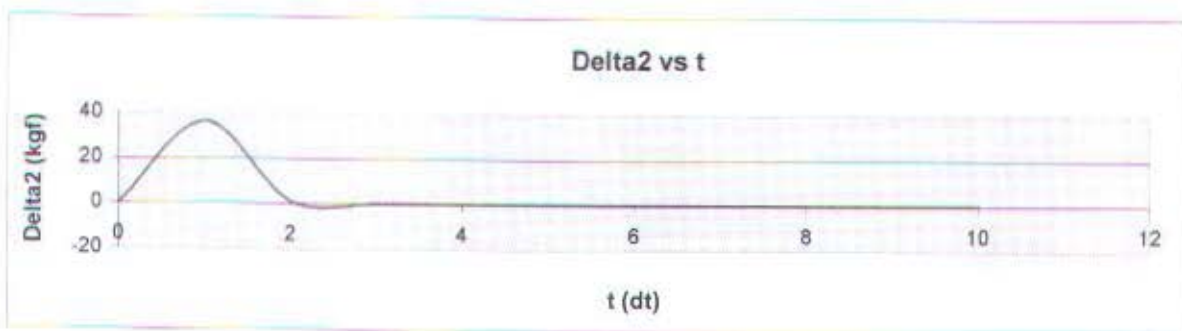
SR (deg)	t (dt)	$\Delta 2$ (kgf)	$r_{SR}$ (m)	$e^{-1.09t}$	$\sin 1.01t$
0	0	0	-	-	-
5	1	36.4	-	-	-
0.109855	2	1.11	0.765	0.113042	0.035248
0.001689	3	0.0173	0.765	0.038006	0.052859
1.18E-05	4	1.20E-04	0.765	0.012778	0.070453
3.44E-08	5	3.51E-07	0.765	0.004296	0.088025
4.05E-11	6	4.13E-10	0.765	0.001444	0.10557
1.87E-14	7	1.89E-13	0.765	0.000486	0.123082
3.29E-18	8	-1.14E-15	0.765	0.000163	0.140556
-7.5E-21	9	-1.17E-15	0.765	5.49E-05	0.157986
-2.9E-21	10	-1.17E-15	0.765	1.85E-05	0.175367

Tab. 5.2 Tabel simulasi terhadap sudut rolling





(a)



(b)



(c)

Graf. 5.2 Grafik simulasi terhadap sudut rolling

(a) Hubungan antara SR vs t

(b) Hubungan antara delta2 vs t

(c) Hubungan antara delta2 vs SR



### 5.1.3 Simulasi Terhadap Sudut Pitching (SPt)

Dalam melakukan simulasi terhadap sudut rolling ini, dilakukan pendekatan dengan menggunakan persamaan umum gerak (Rawson, 1984), seperti berikut ini :

$$\dot{SPt} + 2k_{yy}\omega_0 SPt + \omega_0^2 SP = \Delta 3 \times r_{SPt} \dots\dots\dots (5.3a)$$

dimana :

- $k_{yy} \rightarrow 0.24L \sim 0.26L$  (Rameswar, 1978)

$$k_{yy} = 0.25L = 0.25 \times 2.950 \\ = 0.738 \text{ m}$$

- $I_y = \frac{D}{g} (k_{yy})^2 = \frac{3000}{9.8} (0.738)^2 \\ = 166.73 \text{ kg} \cdot \text{dt}^2 \cdot \text{m}$

- $(R-a) \cong R = \frac{\alpha^2}{14\delta} \times \frac{L^2}{H} \\ = \frac{1^2}{14 \times 1} \times \frac{(2.950)^2}{2} \\ = 0.31 \text{ m}$

- $\omega_0^2 = \frac{D(R-a)}{I_y} \\ = \frac{3000 \times 0.31}{166.73} = 5.58 \text{ (rad / dt)}^2$

- $\omega_0 = 2.36 \text{ rad/dt}$

sehingga persamaan (5.3a) dapat dituliskan sebagai berikut :

$$\dot{SPt} + 3.48 SPt + 5.58 SPt = \Delta 3 \times r_{SPt} \dots\dots\dots (5.3b)$$



dengan mengubah persamaan (5.3b) kedalam fungsi bilangan kompleks, maka didapatkan :

$$\frac{SPt(s)}{\Delta 3 \times r_{SPt}} = \frac{1}{s^2 + 3.48s + 5.58} \dots\dots\dots (5.3c)$$

$$s_{1,2} = \frac{-3.48 \pm \sqrt{12.11 - 22.32}}{2}$$

$$= \frac{-3.48 \pm \sqrt{-10.21}}{2} = -1.74 \pm 1.60j$$

$$\frac{a1(s - \alpha) + a2\beta}{(s - \alpha)^2 + \beta^2} = \frac{a1(s + 1.74) + 1.60a2}{(s + 1.74)^2 + (1.60)^2}$$

$$a1(s + 1.74) + 1.60a2 = 1 \Big|_{s = -1.74 + 1.60j}$$

$$a1(-1.74 + 1.60j + 1.74) + 1.60a2 = 1$$

$$a1 = 0 \quad / \quad a2 = \frac{1}{1.60}$$

persamaan (5.3c), dapat dituiskan kembali menjadi :

$$\frac{SPt(s)}{\Delta 3 \times r_{SPt}} = \frac{1}{(s + 1.74)^2 + (1.60)^2} \dots\dots\dots (5.3d)$$

kemudian persamaan (5.3d) diubah kedalam fungsi waktu yang kemudian akan disimulasikan, sebagai berikut :

$$SPt(t) = \frac{\Delta 3 \times r_{SPt} \times e^{-1.74t} \times \sin 1.60t}{1.60} \dots\dots\dots (5.3e)$$

dimana : -  $\Delta 3$  merupakan hasil dari defuzzifikasi berdasarkan masukan dari SPT

$$- r_{SR} = 1.325 \text{ m}$$

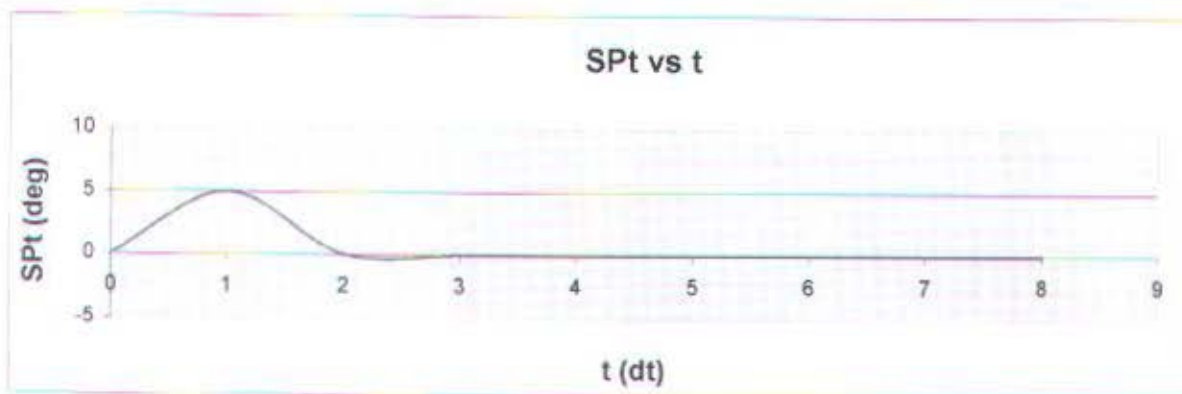
Dengan menggunakan persamaan (5.3e), maka simulasi terhadap sudut pitching didapatkan seperti yang diperlihatkan pada tabel dan grafik 5.3 berikut ini :



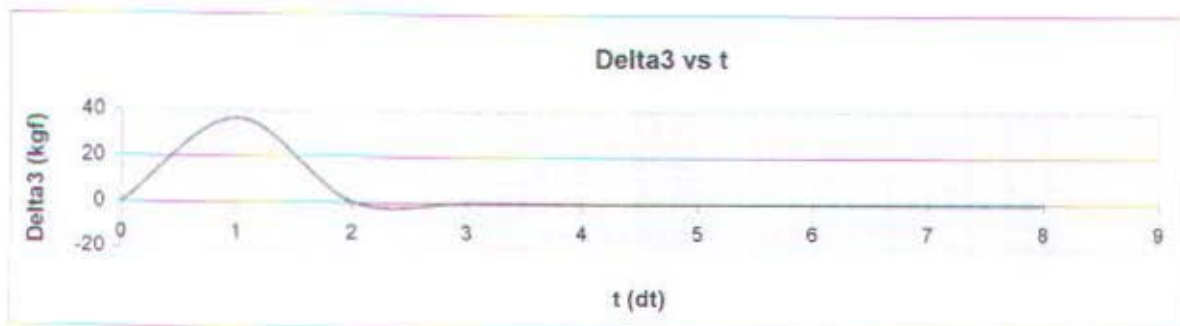


SPt (deg)	t (dt)	$\Delta 3$ (kgf)	$r_{SPt}$ (m)	$e^{-1.74t}$	$\sin 1.60t$
0	0	0	-	-	-
5	1	36.4	-	-	-
0.051839	2	0.528	1.325	0.030807	0.055822
0.000198	3	0.00202	1.325	0.005407	0.083678
1.77E-07	4	1.80E-06	1.325	0.000949	0.111469
3.46E-11	5	3.53E-10	1.325	0.000167	0.139173
1.43E-15	6	1.34E-14	1.325	2.92E-05	0.166769
1.11E-20	7	-1.17E-15	1.325	5.13E-06	0.194234
-1.9E-22	8	-1.17E-15	1.325	9.01E-07	0.221548

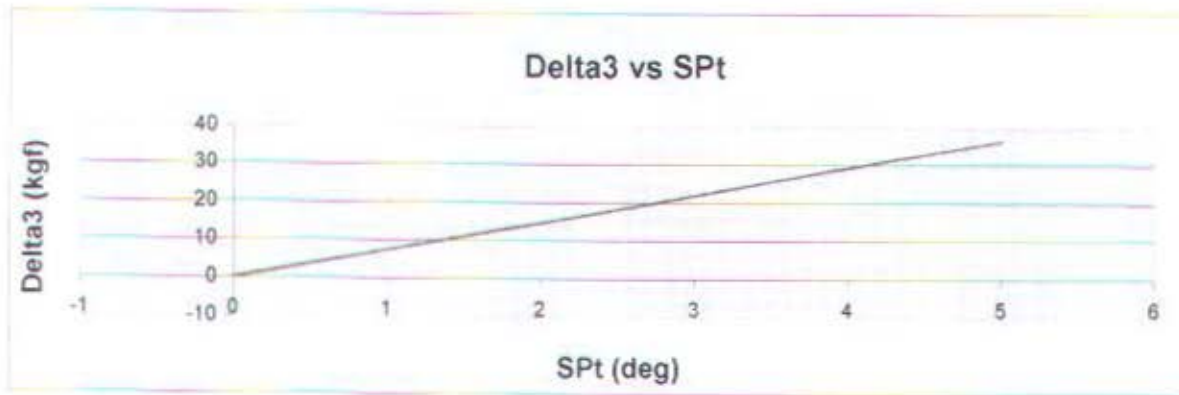
Tab. 5.3 Tabel simulasi terhadap sudut pitching



(a)



(b)



(c)

Graf. 5.3 Grafik simulasi terhadap sudut pitching

- (a) Hubungan antara SPt vs t
- (b) Hubungan antara delta3 vs t
- (c) Hubungan antara delta3 vs SPt

## 5.2 Analisa

Setelah kita melakukan simulasi terhadap setiap variabel masukan dan keluaran yang telah kita tentukan seperti diuraikan diatas, dimana simulasi yang kita lakukan dalam penulisan tugas akhir ini hanya terbatas pada penyajian tampilan dalam bentuk tabel dan grafik, maka langkah selanjutnya yaitu menganalisa hasil yang kita peroleh berdasarkan pada simulasi yang kita telah lakukan.

### 5.2.1 Analisa Terhadap Sudut Posisi

Analisa yang dilakukan disini didasarkan pada simulasi seperti yang diuraikan sebelumnya. Dengan menggunakan persamaan (5.1e), maka kita akan mendapatkan hasil seperti pada tabel dan grafik 5.1. Pada tabel 5.1, kita dapat melihat hasil perhitungan simulasi dengan menggunakan persamaan (5.1e) yang kemudian dari tabel 5.1 tersebut kita dapat membuat hubungan yang ingin kita tampilkan dalam bentuk tampilan grafik.



Pada grafik 5.1a, menunjukkan hubungan antara sudut posisi dengan waktu yang diperlukan untuk mencapai kondisi yang diinginkan (kondisi dimana rov berjalan lurus pada arah belok sekian derajat dari kondisi awalnya). Dengan memperhatikan tampilan grafik 5.1a, jika terjadi variabel masukan sudut posisi sebesar  $10^0$  maka waktu yang diperlukan untuk mencapai kondisi masukan sebesar tersebut yaitu selama detik ke-3 dengan menganggap  $SP = 0$  jika  $SP \leq 10^{-3}$  dan  $\Delta l = 0$  jika  $\Delta l \leq 10^{-2}$ . Jika kita memasukkan variabel masukan sudut posisi sebesar  $0^0$  maka  $\Delta l$  yang dibutuhkan sebesar  $3.3 \times 10^{-15}$  (pada tampilan perhitungan program Matlab), dengan menganggap nilai  $\Delta l$  sebesar itu sama dengan nol, dengan kata lain tidak terjadi  $\Delta l$  (perbedaan thrust). Namun dalam penulisan tugas akhir ini, kita asumsikan bahwa  $SP = 0$  jika  $SP \leq 10^{-3}$  dan  $\Delta l = 0$  jika  $\Delta l \leq 10^{-2}$ . Sehingga dapat dikatakan bahwa jika kita menginginkan rov berbelok kekanan sebesar  $10^0$ , maka waktu yang dibutuhkan oleh rov untuk mencapai kondisi belok sebesar  $10^0$  atau rov sudah berada pada arah belok  $10^0$  dari kondisi awalnya selama detik ke-3, dengan diikuti oleh penurunan perbedaan thrust antara port dan stbd ( $\Delta l$ ) sampai tidak terjadi perbedaan thrust lagi antara port dan stbd ( $\Delta l$ ), sehingga dengan kondisi ini rov sudah berjalan lurus pada arah  $10^0$  dari kondisi semula. Untuk lebih jelasnya dapat dilihat pada tabel dan grafik 5.1 yang tersedia diatas.

Pada grafik 5.1b, menunjukkan hubungan antara  $\Delta l$  (perbedaan thrust) dengan lamanya waktu yang diperlukan untuk mencapai kondisi yang diinginkan berdasarkan pada besarnya masukan yang diterimanya. Dari grafik 5.1b ini kita dapat memperhatikan penurunan dari perbedaan thrust yang terjadi seiring dengan lamanya waktu yang diperlukan untuk memenuhi kondisi yang diinginkan.





Sedangkan pada grafik 5.1c, menunjukkan pada kita hubungan antara sudut posisi dengan delta1, yang mana semakin besar sudut posisi yang kita berikan semakin besar kebutuhan delta1 yang diperlukan. Dengan kata lain dapat kita katakan bahwa semakin besar masukan yang kita berikan semakin besar kebutuhan delta1 yang diperlukan tetapi lama waktu yang dibutuhkan untuk mencapai kondisi yang diinginkan tetap sama yaitu detik ke-3 berapapun besarnya variabel masukan sudut posisi yang terjadi, seperti yang diperlihatkan pada tabel dan grafik 5.4.

SP (deg)	t (dt)	$\Delta 1$ (kgf)	$r_{SP}$ (m)	$e^{-2.21t}$	$\sin 1.71t$
0	0	0	-	-	-
5	1	75	-	-	-
0.022828	2	0.513	0.725	0.012034	0.059655
2.57E-05	3	0.000574	0.725	0.00132	0.089416
4.2E-09	4	9.38E-08	0.725	0.000145	0.119097
9.39E-14	5	2.10E-12	0.725	1.59E-05	0.148672
2.76E-19	6	3.30E-15	0.725	1.74E-06	0.178115
5.55E-23	7	3.30E-15	0.725	1.91E-07	0.2074

(a)

SP (deg)	t (dt)	$\Delta 1$ (kgf)	$r_{SP}$ (m)	$e^{-2.21t}$	$\sin 1.71t$
0	0	0	-	-	-
10	1	150	-	-	-
0.045656	2	1.02	0.725	0.012034	0.059655
5.1E-05	3	0.00114	0.725	0.00132	0.089416
8.34E-09	4	1.86E-07	0.725	0.000145	0.119097
1.86E-13	5	4.16E-12	0.725	1.59E-05	0.148672



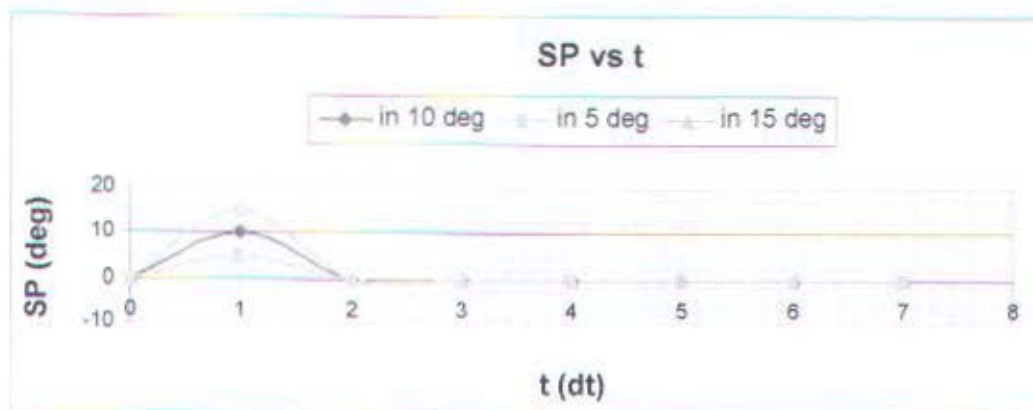
5.48E-19	6	3.30E-15	0.725	1.74E-06	0.178115
5.55E-23	7	3.30E-15	0.725	1.91E-07	0.2074

(b)

SP (deg)	t (dt)	$\Delta 1$ (kgf)	$r_{SP}$ (m)	$e^{-2.21t}$	$\sin 1.71t$
0	0	0	-	-	-
15	1	215	-	-	-
0.06544	2	1.44	0.725	0.012034	0.059655
7.21E-05	3	0.00161	0.725	0.00132	0.089416
1.18E-08	4	2.64E-07	0.725	0.000145	0.119097
2.64E-13	5	5.90E-12	0.725	1.59E-05	0.148672
7.77E-19	6	3.30E-15	0.725	1.74E-06	0.178115
5.55E-23	7	3.30E-15	0.725	1.91E-07	0.2074

(c)

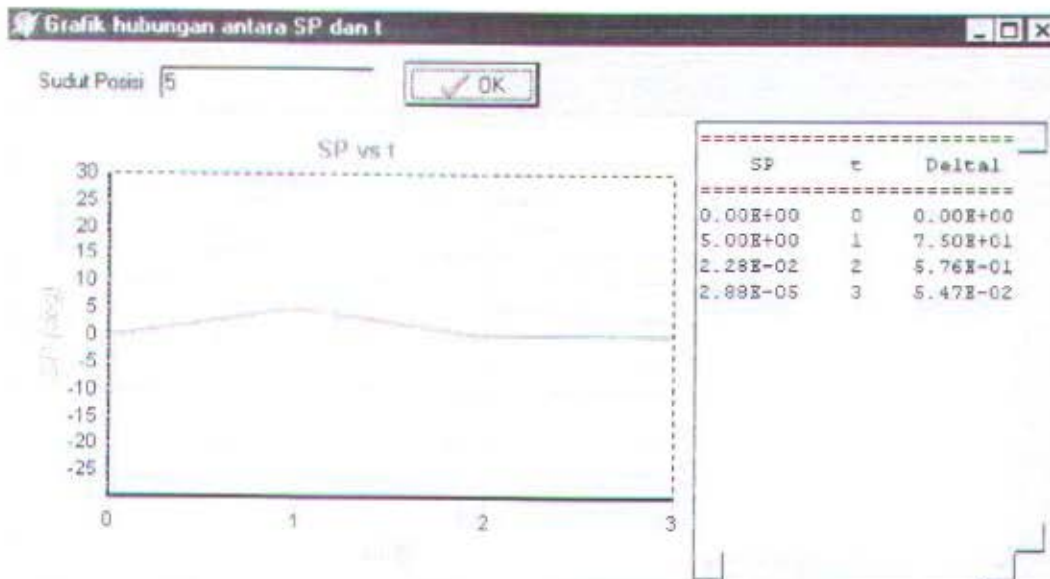
Tab. 5.4 Tabel simulasi terhadap variasi sudut posisi

(a) Sudut posisi dengan input  $5^0$ (b) Sudut posisi dengan input  $10^0$ (c) Sudut posisi dengan input  $15^0$ 

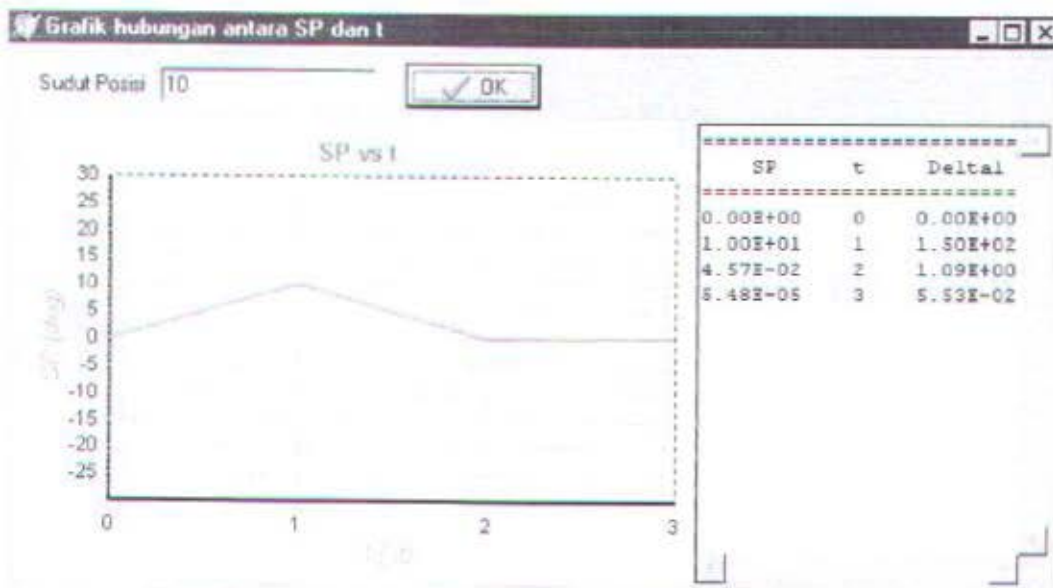
Graf. 5.4 Grafik simulasi terhadap variasi sudut posisi



Pada gambar 5.7 juga diperlihatkan bahwa berapapun besar variabel masukan sudut posisi yang ada, lama waktu yang dibutuhkan akan tetap sama yaitu pada detik ke-3. Hal ini dikarenakan semakin besarnya variabel masukan sudut posisi yang terjadi diikuti dengan peningkatan kebutuhan perbedaan thrust ( $\Delta t$ ) yang timbul, seperti yang diperlihatkan pada grafik 5.1c.

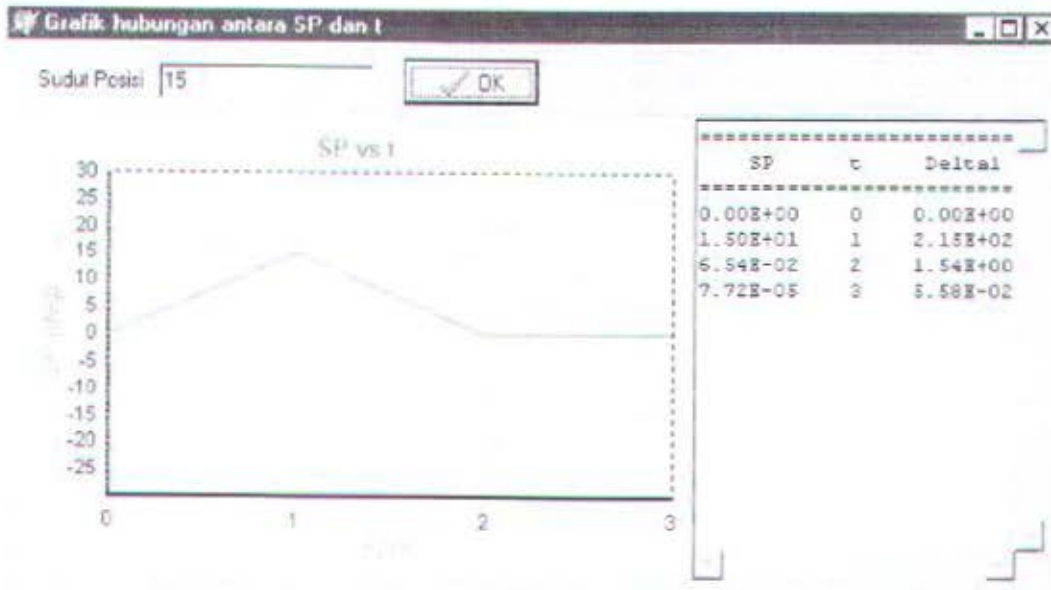


(a)



(b)





(c)

Gb. 5.7 Grafik &amp; tabel hasil simulasi sudut posisi – program delphi

(a) sudut posisi  $5^0$  (b) sudut posisi  $10^0$  (c) sudut posisi  $15^0$ 

### 5.2.2 Analisa Terhadap Sudut Rolling

Analisa yang dilakukan disini didasarkan pada simulasi seperti yang diuraikan sebelumnya. Dengan menggunakan persamaan (5.2e), maka kita akan mendapatkan hasil seperti pada tabel dan grafik 5.2. Pada tabel 5.2, kita dapat melihat hasil perhitungan simulasi dengan menggunakan persamaan (5.2e) yang kemudian dari tabel 5.2 tersebut kita dapat membuat hubungan yang ingin kita tampilkan dalam bentuk tampilan grafik.

Pada grafik 5.2a, menunjukkan hubungan antara sudut rolling dengan waktu yang diperlukan untuk mencapai kondisi yang diinginkan (kondisi balik pada posisi semula dimana rov tidak mengalami kondisi rolling lagi). Dengan memperhatikan tampilan grafik 5.2a, jika terjadi variabel masukan sudut rolling sebesar  $5^0$  maka waktu yang diperlukan untuk mencapai kondisi masukan sebesar tersebut yaitu selama detik ke-4 dengan menganggap  $SR = 0$  jika  $SR \leq 10^{-3}$  dan



$\Delta 2 = 0$  jika  $\Delta 2 \leq 10^{-2}$ . Jika kita memasukkan variabel masukan sudut rolling sebesar  $0^\circ$  maka  $\Delta 2$  yang dibutuhkan sebesar  $-1.17 \times 10^{-15}$  (pada tampilan perhitungan program Matlab), dengan menganggap nilai  $\Delta 2$  sebesar itu sama dengan nol, dengan kata lain tidak terjadi  $\Delta 2$  (perbedaan thrust). Namun dalam penulisan tugas akhir ini, kita asumsikan bahwa  $SR = 0$  jika  $SR \leq 10^{-3}$  dan  $\Delta 2 = 0$  jika  $\Delta 2 \leq 10^{-2}$ . Sehingga dapat dikatakan bahwa jika kita menginginkan roV kembali pada kondisi semula setelah menerima masukan sudut rolling sebesar  $5^\circ$ , maka waktu yang dibutuhkan oleh roV untuk mencapai kondisi semula sehingga roV tidak mengalami kondisi rolling lagi selama detik ke-4, dengan diikuti oleh penurunan perbedaan thrust antara port dan stbd ( $\Delta 2$ ) sampai tidak terjadi perbedaan thrust lagi antara port dan stbd ( $\Delta 2$ ), sehingga dengan kondisi ini roV sudah kembali pada kondisi semula tanpa mengalami rolling. Untuk lebih jelasnya dapat dilihat pada tabel dan grafik 5.2 yang tersedia diatas.

Pada grafik 5.2b, menunjukkan hubungan antara  $\Delta 2$  (perbedaan thrust) dengan lamanya waktu yang diperlukan untuk mencapai kondisi yang diinginkan berdasarkan pada besarnya masukan yang diterimanya. Dari grafik 5.2b ini kita dapat memperhatikan penurunan dari perbedaan thrust yang terjadi seiring dengan lamanya waktu yang diperlukan untuk memenuhi kondisi yang diinginkan. Sedangkan pada grafik 5.2c, menunjukkan pada kita hubungan antara sudut rolling dengan  $\Delta 2$ , yang mana semakin besar sudut rolling yang kita berikan semakin besar kebutuhan  $\Delta 2$  yang diperlukan. Dengan kata lain dapat kita katakan bahwa semakin besar masukan yang kita berikan semakin besar kebutuhan  $\Delta 2$  yang diperlukan tetapi lama waktu yang dibutuhkan untuk mencapai kondisi yang diinginkan tetap sama yaitu detik ke-4 berapapun besarnya





variabel masukan sudut rolling yang terjadi, seperti yang diperlihatkan pada tabel dan grafik 5.5.

SR (deg)	t (dt)	$\Delta Z$ (kgf)	$r_{SR}$ (m)	$e^{-1.09t}$	$\sin 1.01t$
0	0	0	-	-	-
3	1	24.2	-	-	-
0.073035	2	0.741	0.765	0.113042	0.035248
0.001128	3	0.0112	0.765	0.038006	0.052859
7.64E-06	4	7.79E-05	0.765	0.012778	0.070453
2.23E-08	5	2.27E-07	0.765	0.004296	0.088025
2.62E-11	6	2.67E-10	0.765	0.001444	0.10557
1.21E-14	7	1.22E-13	0.765	0.000486	0.123082
2.12E-18	8	-1.15E-15	0.765	0.000163	0.140556
-7.6E-21	9	-1.17E-15	0.765	5.49E-05	0.157986
-2.9E-21	10	-1.17E-15	0.765	1.85E-05	0.175367

(a)

SR (deg)	t (dt)	$\Delta Z$ (kgf)	$r_{SR}$ (m)	$e^{-1.09t}$	$\sin 1.01t$
0	0	0	-	-	-
5	1	36.4	-	-	-
0.109855	2	1.11	0.765	0.113042	0.035248
0.001689	3	0.0173	0.765	0.038006	0.052859
1.18E-05	4	1.20E-04	0.765	0.012778	0.070453
3.44E-08	5	3.51E-07	0.765	0.004296	0.088025
4.05E-11	6	4.13E-10	0.765	0.001444	0.10557
1.87E-14	7	1.89E-13	0.765	0.000486	0.123082
3.29E-18	8	-1.14E-15	0.765	0.000163	0.140556





-7.5E-21	9	-1.17E-15	0.765	5.49E-05	0.157986
-2.9E-21	10	-1.17E-15	0.765	1.85E-05	0.175367

(b)

SR (deg)	t (dt)	$\Delta z$ (kgf)	$r_{SR}$ (m)	$e^{-1.09t}$	$\sin 1.01t$
0	0	0	-	-	-
7	1	47.3	-	-	-
0.142751	2	1.41	0.765	0.113042	0.035248
0.002146	3	0.0214	0.765	0.038006	0.052859
1.46E-05	4	1.49E-04	0.765	0.012778	0.070453
4.27E-08	5	4.35E-07	0.765	0.004296	0.088025
5.02E-11	6	5.12E-10	0.765	0.001444	0.10557
2.32E-14	7	2.35E-13	0.765	0.000486	0.123082
4.09E-18	8	-1.13E-15	0.765	0.000163	0.140556
-7.4E-21	9	-1.17E-15	0.765	5.49E-05	0.157986
-2.9E-21	10	-1.17E-15	0.765	1.85E-05	0.175367

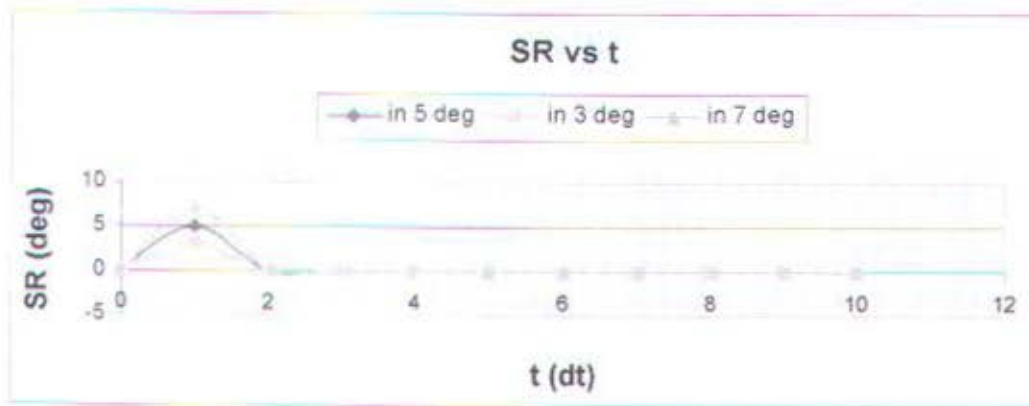
(c)

Tab. 5.5 Tabel simulasi terhadap variasi sudut rolling

(a) Sudut rolling dengan input  $3^\circ$

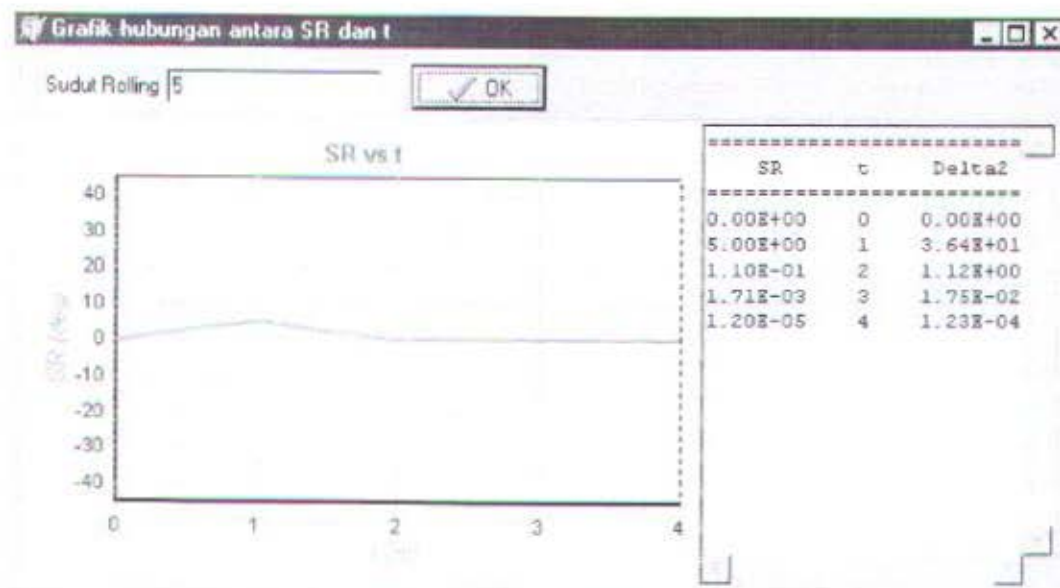
(b) Sudut rolling dengan input  $5^\circ$

(c) Sudut rolling dengan input  $7^\circ$



Graf. 5.5 Grafik simulasi terhadap variasi sudut rolling

Pada gambar 5.8 juga diperlihatkan bahwa berapapun besar variabel masukan sudut rolling yang terjadi, lama waktu yang dibutuhkan akan tetap sama yaitu pada detik ke-4. Hal ini dikarenakan semakin besarnya variabel masukan sudut rolling yang terjadi diikuti dengan peningkatan kebutuhan perbedaan thrust ( $\Delta_2$ ) yang timbul, seperti yang diperlihatkan pada grafik 5.2c.



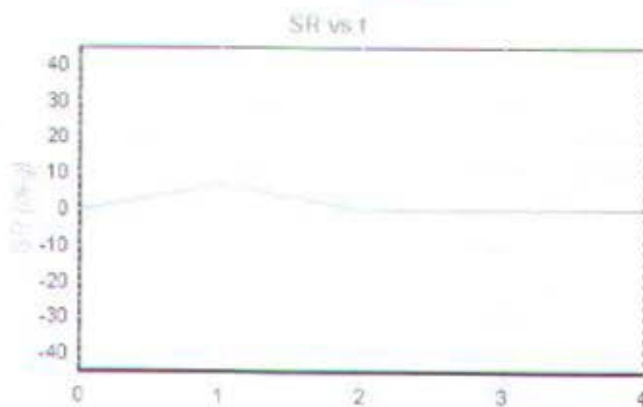
(a)



Grafik hubungan antara SR dan t

Sudut Rolling 7

OK



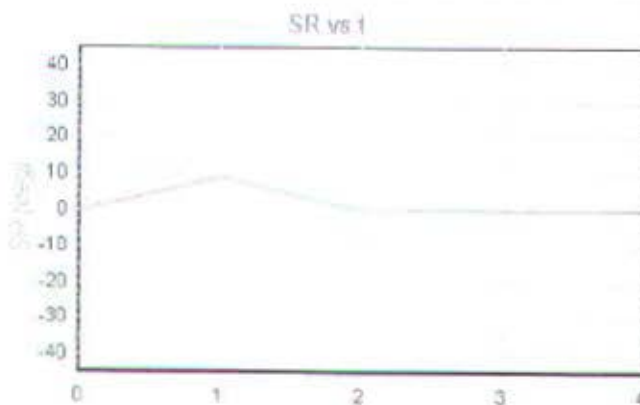
SR	t	Delta2
0.00E+00	0	0.00E+00
7.00E+00	1	4.74E+01
1.43E-01	2	1.46E+00
2.22E-03	3	2.28E-02
1.55E-05	4	1.60E-04

(b)

Grafik hubungan antara SR dan t

Sudut Rolling 9

OK



SR	t	Delta2
0.00E+00	0	0.00E+00
9.00E+00	1	5.39E+01
1.63E-01	2	1.65E+00
2.51E-03	3	2.58E-02
1.76E-05	4	1.81E-04

(c)

Gb. 5.8 Grafik &amp; tabel hasil simulasi sudut rolling – program delphi

(a) sudut rolling  $5^\circ$  (b) sudut rolling  $7^\circ$  (c) sudut rolling  $9^\circ$ 

### 5.2.3 Analisa Terhadap Sudut Pitching

Analisa yang dilakukan disini didasarkan pada simulasi seperti yang diuraikan sebelumnya. Dengan menggunakan persamaan (5.3e), maka kita akan mendapatkan hasil seperti pada tabel dan grafik 5.3. Pada tabel 5.3, kita dapat





melihat hasil perhitungan simulasi dengan menggunakan persamaan (5.3e) yang kemudian dari tabel 5.3 tersebut kita dapat membuat hubungan yang ingin kita tampilkan dalam bentuk tampilan grafik.

Pada grafik 5.3a, menunjukkan hubungan antara sudut pitching dengan waktu yang diperlukan untuk mencapai kondisi yang diinginkan (kondisi balik pada posisi semula dimana rov tidak mengalami kondisi pitching lagi). Dengan memperhatikan tampilan grafik 5.3a, jika terjadi variabel masukan sudut pitching sebesar  $5^{\circ}$  maka waktu yang diperlukan untuk mencapai kondisi masukan sebesar tersebut yaitu selama detik ke-3 dengan menganggap  $SP = 0$  jika  $SP \leq 10^{-3}$  dan  $\Delta T = 0$  jika  $\Delta T \leq 10^{-2}$ . Jika kita memasukkan variabel masukan sudut pitching sebesar  $0^{\circ}$  maka  $\Delta T$  yang dibutuhkan sebesar  $-1.17 \times 10^{-15}$  (pada tampilan perhitungan program Matlab), dengan menganggap nilai  $\Delta T$  sebesar itu sama dengan nol, dengan kata lain tidak terjadi  $\Delta T$  (perbedaan thrust). Namun dalam penulisan tugas akhir ini, kita asumsikan bahwa  $SP = 0$  jika  $SP \leq 10^{-3}$  dan  $\Delta T = 0$  jika  $\Delta T \leq 10^{-2}$ . Sehingga dapat dikatakan bahwa jika kita menginginkan rov kembali pada kondisi semula setelah menerima masukan sudut pitching sebesar  $5^{\circ}$ , maka waktu yang dibutuhkan oleh rov untuk mencapai kondisi semula sehingga rov tidak mengalami kondisi pitching lagi selama detik ke-3, dengan diikuti oleh penurunan perbedaan thrust antara depan dan belakang ( $\Delta T$ ) sampai tidak terjadi perbedaan thrust lagi antara depan dan belakang ( $\Delta T$ ), sehingga dengan kondisi ini rov sudah kembali pada kondisi semula tanpa mengalami pitching. Untuk lebih jelasnya dapat dilihat pada tabel dan grafik 5.3 yang tersedia diatas.



Pada grafik 5.3b, menunjukkan hubungan antara  $\Delta 3$  (perbedaan thrust) dengan lamanya waktu yang diperlukan untuk mencapai kondisi yang diinginkan berdasarkan pada besarnya masukan yang diterimanya. Dari grafik 5.3b ini kita dapat memperhatikan penurunan dari perbedaan thrust yang terjadi seiring dengan lamanya waktu yang diperlukan untuk memenuhi kondisi yang diinginkan. Sedangkan pada grafik 5.3c, menunjukkan pada kita hubungan antara sudut pitching dengan  $\Delta 3$ , yang mana semakin besar sudut pitching yang kita berikan semakin besar kebutuhan  $\Delta 3$  yang diperlukan. Dengan kata lain dapat kita katakan bahwa semakin besar masukan yang kita berikan semakin besar kebutuhan  $\Delta 3$  yang diperlukan tetapi lama waktu yang dibutuhkan untuk mencapai kondisi yang diinginkan tetap sama yaitu detik ke-3 berapapun besarnya variabel masukan sudut pitching yang terjadi, seperti yang diperlihatkan pada tabel dan grafik 5.6.

SPT (deg)	t (dt)	$\Delta 3$ (kgf)	$r_{SPT}$ (m)	$e^{-1.74t}$	$\sin 1.60t$
0	0	0	-	-	-
3	1	24.2	-	-	-
0.034464	2	0.346	1.325	0.030807	0.055822
0.00013	3	0.00132	1.325	0.005407	0.083678
1.16E-07	4	1.18E-06	1.325	0.000949	0.111469
2.27E-11	5	2.31E-10	1.325	0.000167	0.139173
9.33E-16	6	8.34E-15	1.325	2.92E-05	0.166769
6.88E-21	7	-1.17E-15	1.325	5.13E-06	0.194234
-1.9E-22	8	-1.17E-15	1.325	9.01E-07	0.221548

(a)





SPl	t	$\Delta 3$	$r_{SPl}$	$e^{-1.74t}$	$\sin 1.60t$
(deg)	(dt)	(kgf)	(m)		
0	0	0	-	-	-
5	1	36.4	-	-	-
0.051839	2	0.528	1.325	0.030807	0.055822
0.000198	3	0.00202	1.325	0.005407	0.083678
1.77E-07	4	1.80E-06	1.325	0.000949	0.111469
3.46E-11	5	3.53E-10	1.325	0.000167	0.139173
1.43E-15	6	1.34E-14	1.325	2.92E-05	0.166769
1.11E-20	7	-1.17E-15	1.325	5.13E-06	0.194234
-1.9E-22	8	-1.17E-15	1.325	9.01E-07	0.221548

(b)

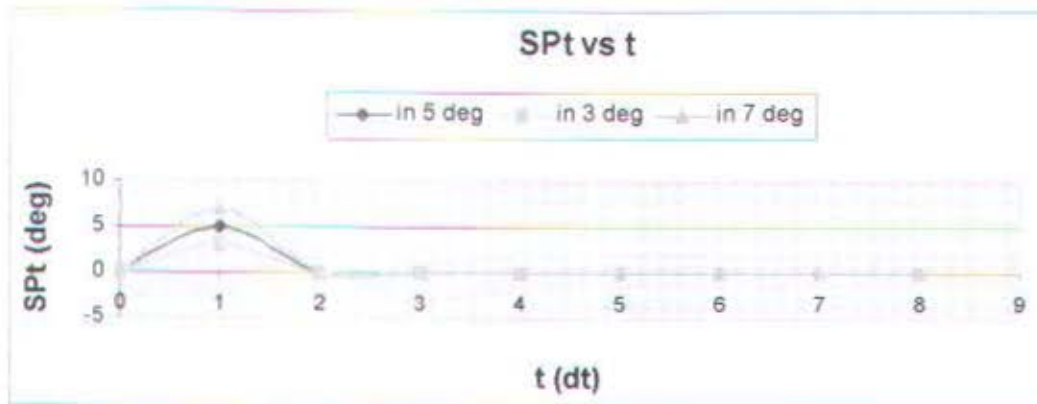
SPl	t	$\Delta 3$	$r_{SPl}$	$e^{-1.74t}$	$\sin 1.60t$
(deg)	(dt)	(kgf)	(m)		
0	0	0	-	-	-
7	1	47.3	-	-	-
0.067362	2	0.68	1.325	0.030807	0.055822
0.000255	3	0.0026	1.325	0.005407	0.083678
2.28E-07	4	2.32E-06	1.325	0.000949	0.111469
4.45E-11	5	4.54E-10	1.325	0.000167	0.139173
1.83E-15	6	1.75E-14	1.325	2.92E-05	0.166769
1.44E-20	7	-1.17E-15	1.325	5.13E-06	0.194234
-1.9E-22	8	-1.17E-15	1.325	9.01E-07	0.221548

(c)

Tab. 5.6 Tabel simulasi terhadap variasi sudut pitching

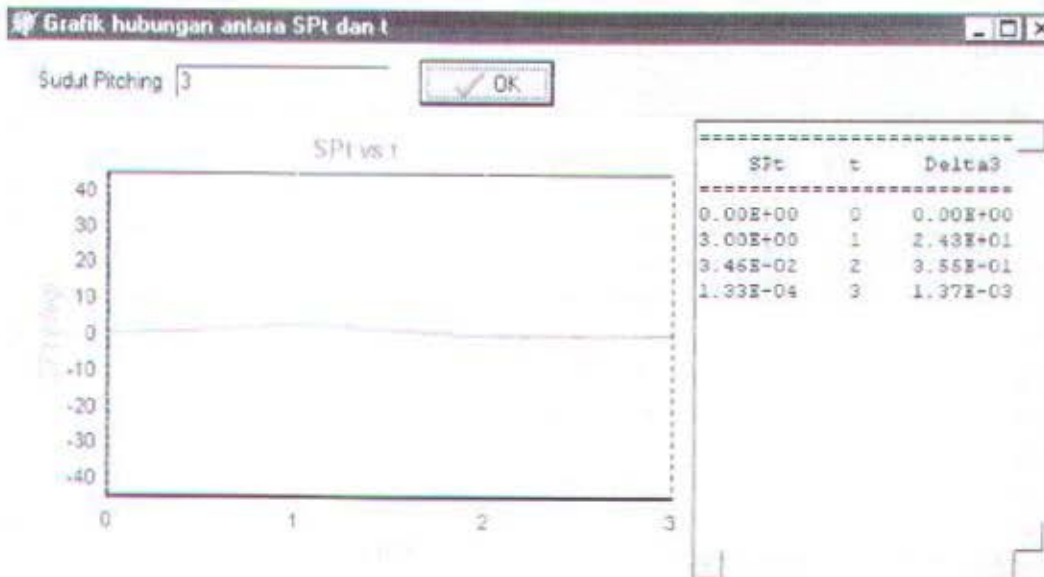
(a) Sudut pitching dengan input  $3^0$ (b) Sudut pitching dengan input  $5^0$



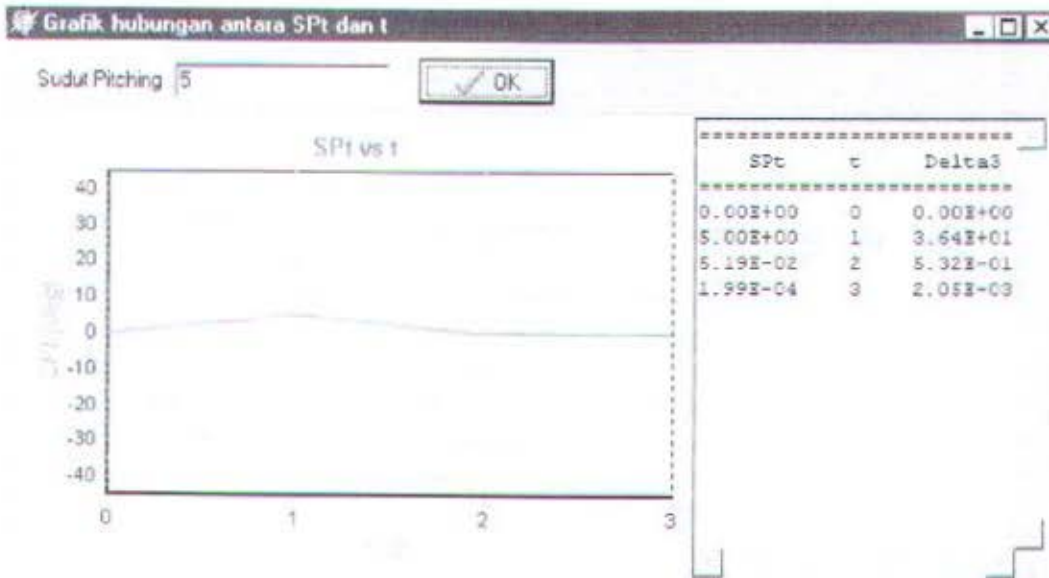
(c) Sudut pitching dengan input  $7^\circ$ 

Graf. 5.6 Grafik simulasi terhadap variasi sudut pitching

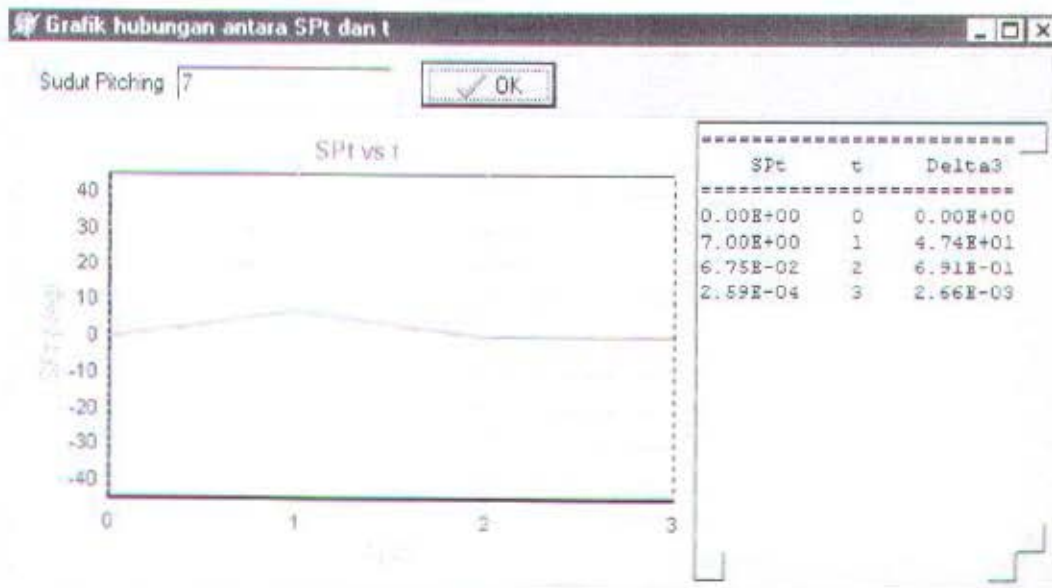
Pada gambar 5.9 juga diperlihatkan bahwa berapapun besar variabel masukan sudut pitching yang terjadi, lama waktu yang dibutuhkan akan tetap sama yaitu pada detik ke-3. Hal ini dikarenakan semakin besarnya variabel masukan sudut pitching yang terjadi diikuti dengan peningkatan kebutuhan perbedaan thrust ( $\Delta 3$ ) yang timbul, seperti yang diperlihatkan pada grafik 5.3c.



(a)



(b)



(c)

Gb. 5.9 Grafik &amp; tabel hasil simulasi sudut pitching – program delphi

(a) sudut pitching  $3^{\circ}$  (b) sudut pitching  $5^{\circ}$  (c) sudut pitching  $7^{\circ}$



## BAB VI

### PENUTUP

#### 6.1 Kesimpulan

Dari uraian penulisan tugas akhir diatas, dapat disimpulkan bahwa :

- 1) Dengan menerapkan/merancang sistem kendali logika fuzzy maka akan diperoleh efektifitas kestabilan gerak rov pada waktu yang relatif singkat.
- 2) Dengan harga variabel masukan sudut posisi sebesar  $10^0$ , maka waktu yang diperlukan untuk mencapai kondisi yang diinginkan/berjalan lurus dalam arah belok  $10^0$  dari kondisi awal selama detik ke-3.
- 3) Dengan harga variabel masukan sudut rolling sebesar  $5^0$ , maka waktu yang diperlukan untuk mencapai kondisi yang diinginkan/tidak terjadi kondisi rolling lagi selama detik ke-4.
- 4) Dengan harga variabel masukan sudut pitching sebesar  $5^0$ , maka waktu yang diperlukan untuk mencapai kondisi yang diinginkan/ tidak terjadi kondisi pitching lagi selama detik ke-3.
- 5) Semakin besar variabel masukan yang diterima, baik untuk variabel masukan sudut posisi, sudut rolling, dan sudut pitching, semakin besar kebutuhan perbedaan thrust yang diperlukan tetapi lama waktu yang diperlukan untuk mencapai kondisi yang diinginkan tetap sama untuk tiap-tiap variabel masukan.

#### 6.2 Saran

Dalam penulisan tugas akhir ini, masih banyak asumsi, kekurangan dan hal-hal yang tidak dibahas, seperti aplikasi pengontrolan untuk gerak rov dalam 6





derajat kebebasan, aplikasi terhadap servovalve, asumsi range pemetaan variabel masukan dan keluaran yang hanya didasarkan pada asumsi penulis tidak pada kondisi sebenarnya, dan masih banyak lagi.

Untuk itulah penulis mengharapkan agar bagi mereka yang berkeinginan untuk mengambil topik yang berhubungan dengan apa yang telah dikerjakan penulis, sekiranya dapat lebih menyempurnakan apa-apa yang telah penulis lakukan kurang lebihnya, sehingga bisa memberikan gambaran yang lebih bermanfaat dikemudian hari.



## DAFTAR PUSTAKA

1. Bhattacharyya, Rameswar [1978], **"Dynamics of Marine Vehicles"**, A Wiley-Interscience Publishing.
2. Blagoveshchensky, S.N. [1988], **"Theory of Ship Motions"**, Volume 1, Dover Publishing, Inc. New York.
3. Brown, Martin., Harris, Chris [1994], **"Neurofuzzy Adaptive Modelling & Control"**, Prentice Hall International, Inc.
4. Conte G., dkk. [2002], **"An Automatic Guidance System for a Small Work-Class ROV"**, University of Ancona. Department of Electronics and Automation, Via Brece Bianche - 60131, Ancona – Italy.
5. Goodfellow Associates Ltd. [1996], **"Applications of Subsea Systems"**, Pennwell Publishing Company, Tulsa, Oklahoma.
6. Huda, M. Chairul [1997], **"Perancangan Sistem Kendali Logika Fuzzy Pada Sistem Penukar Panas Untuk Pendingin Minyak Pelumas di PT. PLN PJB II Sektor Gresik"**, Tugas Akhir S-1, ITS.
7. Huda, Midkhol [1997], **"Penerapan Kendali Logika Fuzzy – PI Pada Sistem Penukar Panas"**, Tugas Akhir S-1, ITS.
8. Kosko, Bart [1997], **"Fuzzy Engineering"**, Prentice Hall International, Inc.
9. Michel, Drew [2002], **"ROV Educational Materials"**, ROV Committee.
10. Murtedjo, Ir. M, ME. [1985], **"Teori Gerak Kapal I"**, Jurusan Teknik Perkapalan, FTK – ITS.
11. Negnevitsky, Michael [ ], **"Fuzzy Expert Systems"**, School of Engineering, University of Tasmania.



12. Pedrycz, Witold [1996], **"Fuzzy Modelling Paradigms and Practice"**, Kluwer Academic Publishers.
13. Rawson, K.J., Tupper, F.C. [1984], **"Basic Ship Theory"**, Volume 2, Longman Group Limited.
14. Santoso, Albertus Joko [1996], **"Aplikasi Pengendali Logika Fuzzy Pada Industri"**, Vasthu Edisi Khusus.
15. Tomsovic, K., Chow, M.Y. [2000], **"Tutorial on Fuzzy Logic Applications in Power Systems"**, Mathworks, Inc.
16. Zadeh, L.A. [1995], **"Fuzzy Logic Toolbox for Use With Matlab"**, Version 2, Mathworks, Inc.
17. [www.rov.net](http://www.rov.net)
18. [daveandjill@neteze.com](mailto:daveandjill@neteze.com)
19. [www.hydrovision.co.uk](http://www.hydrovision.co.uk)



## LAMPIRAN A :

Listing Program Control (Gb. 5.7 - 5.9)

```
control sudut posisi
```

```
Ufuzzy1;
```

```
interface
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,  
Dialogs, TeEngine, Series, TeeProcs, Chart, StdCtrls, Buttons, ExtCtrls;
```

```
TFSPvst = class(TForm)  
  Panel1: TPanel;  
  Panel2: TPanel;  
  Panel3: TPanel;  
  Edit1: TEdit;  
  BitBtn1: TBitBtn;  
  Chart1: TChart;  
  Series1: TFastLineSeries;  
  Edit2: TEdit;  
  Series2: TFastLineSeries;  
  Memo1: TMemo;  
  procedure BitBtn1Click(Sender: TObject);  
private  
  { Private declarations }  
public  
  { Public declarations }  
end;
```

```
TFSPvst: TFSPvst;
```

```
implementation
```

```
Form1.dfm;
```

```
procedure TFSPvst.BitBtn1Click(Sender: TObject);  
  sudut, x1, x2, x3, x4, x5, x6, y1, y2, y3, y4, y5, y6, range : real;  
  daerah : Char;  
  besar, kecil : real;  
  Delta, ttk1, ttk2, hasil, atas, bawah : real;  
  t : real;  
  FileR : TextFile;
```

# Ufuzzzy1.pas

```

abs(StrToFloat(Edit1.Text))>30) then
messageDlg('Input melebihi range -30>= input <=30....!', mtInformati
mbOk, 0)
n
signFile(FileR, 'c:\joko.txt');
write(FileR);
writeln(FileR, '=====');
writeln(FileR, '      SP      t      Delta1 ');
writeln(FileR, '=====');
          0.00E+00      1      0.00E+00
writeln(FileR, FormatFloat('0.00E+00', 0) + '      '+FormatFloat('0', t) + '
+FormatFloat('0.00E+00', 0));
ries2.Clear;
=0;
dut:=0;
ries2.AddXY(t, sudut, '', clTeeColor);
dut := strtofloat(Edit1.Text);
=1;
ile true do
gin
Series2.AddXY(t, sudut, '', clTeeColor);
// awal looping
ttk1 := 0;
ttk2 := 0;
if (sudut>=-30) and (sudut<=-20) then
begin
daerah := 'A';
end
else if (sudut>-20) and (sudut<=-15) then
begin
daerah := 'B';
x1:=-20; y1:=0;
x2:=-10; y2:=1;
ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

x3:=-10; y3:=0;
x4:=-20; y4:=1;
ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=-15) then
begin
daerah := 'C'
end
else if (sudut>-15) and (sudut<-10) then
begin
daerah := 'D';

```



Ufuzzzy1.pas

```

x1:=-20; y1:=0;
x2:=-10; y2:=1;
ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

x3:=-10; y3:=0;
x4:=-20; y4:=1;
ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=-10) then
begin
    daerah := 'E'
end
else if (sudut>-10) and (sudut<-5) then
begin
    daerah := 'F';
    x1:=-10; y1:=0;
    x2:=0 ; y2:=1;
    ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

    x3:=0 ; y3:=0;
    x4:=-10; y4:=1;
    ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=-5) then
begin
    daerah := 'G'
end
else if (sudut>-5) and (sudut<0) then
begin
    daerah := 'H';
    x1:=-10; y1:=0;
    x2:=0 ; y2:=1;
    ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

    x3:=0 ; y3:=0;
    x4:=-10; y4:=1;
    ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=0) then
begin
    daerah := 'I'
end
else if (sudut>0) and (sudut<5) then
begin
    daerah := 'J';
    x1:=0 ; y1:=0;
    x2:=10 ; y2:=1;

```

Ufuzzy1.pas

```

ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

x3:=10 ; y3:=0;
x4:=0 ; y4:=1;
ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=5) then
begin
    daerah := 'K'
end
else if (sudut>5) and (sudut<10) then
begin
    daerah := 'L';
    x1:=0 ; y1:=0;
    x2:=10 ; y2:=1;
    ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

    x3:=10 ; y3:=0;
    x4:=0 ; y4:=1;
    ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=10) then
begin
    daerah := 'M'
end
else if (sudut>10) and (sudut<15) then
begin
    daerah := 'N';
    x1:=10 ; y1:=0;
    x2:=20 ; y2:=1;
    ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

    x3:=20 ; y3:=0;
    x4:=10 ; y4:=1;
    ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=15) then
begin
    daerah := 'O'
end
else if (sudut>15) and (sudut<20) then
begin
    daerah := 'P';
    x1:=10 ; y1:=0;
    x2:=20 ; y2:=1;
    ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

```

# Ufuzzy1.pas

```

x3:=20 ; y3:=0;
x4:=10 ; y4:=1;
ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut>=20) and (sudut<=30) then
begin
    daerah := 'Q'
end;

if ttk1>ttk2 then
begin
    besar := ttk1;
    kecil := ttk2;
end
else
begin
    besar := ttk2;
    kecil := ttk1;
end;

case daerah of
'A' : begin
    delta := -304;
end;
'B' : begin
    range:=-390;
    atas:=0;
    bawah:= 0;
    x1:=-150; y1:=0;
    x2:=-300; y2 :=1;
    x3:=0 ; y3:=0;
    x4:=-150; y4 :=1;
    Series1.Clear;
    while true do
    begin
        if (range>=-390) and (range<=-300) then hasil:=besar
        else if (range>-300) and (range<=-150) then
            begin
                hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);
                if hasil>besar then hasil:=besar;
                if hasil<kecil then hasil:=kecil;
            end
        else if (range>-150) and (range<=0) then
            begin
                hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);
                if hasil>kecil then hasil:=kecil;
            end;
    end;
end;

```



```

                                Ufuzzy1.pas
    atas := atas-(hasil*range);
    bawah := bawah+hasil;
    Series1.AddXY(range,hasil,'',clTeeColor);
    range := range+8;
    if range>0 then break;
end;
    delta := atas/bawah;
end;
'C' : begin
    delta := -215;
end;
'D' : begin
    range:=-390;
    atas:=0;
    bawah:= 0;
    x1:=-300; y1:=0;
    x2:=-150; y2 :=1;
    x3:=0    ; y3:=0;
    x4:=-150; y4 :=1;
    Series1.Clear;
    while true do
    begin
        if (range>=-390) and (range<=-300) then hasil:=kecil
        else if (range>-300) and (range<=-150) then
            begin
                hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);
                if hasil>besar then hasil:=besar;
                if hasil<kecil then hasil:=kecil;
            end
        else if (range>-150) and (range<=0) then
            begin
                hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);
                if hasil>besar then hasil:=besar;
            end;
        atas := atas+(hasil*range);
        bawah := bawah+hasil;
        Series1.AddXY(range,hasil,'',clTeeColor);
        range := range+8;
        if range>0 then break;
    end;
    delta := atas/bawah;
end;
'E' : begin
    delta := -150;
end;
F' : begin
    range:=-300;

```

# Ufuzzy1.pas

```

atas:=0;
bawah:= 0;
x1:=-300; y1:=0;
x2:=-150; y2 :=1;
x3:=0    ; y3:=0;
x4:=-150; y4 :=1;
x5:=150  ; y5:=0;
x6:=0    ; y6 :=1;
  Series1.Clear;
while true do
begin
  if (range>=-300) and (range<=-150) then
  begin
    hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);
    if hasil>besar then hasil:=besar;
  end
  else if (range>-150) and (range<=0) then
  begin
    hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);
    if hasil>besar then hasil:=besar;
    if hasil<kecil then hasil:=kecil;
  end
  else if (range>0) and (range<=150) then
  begin
    hasil := (y6*(range-x5)-y5*(range-x6))/(x6-x5);
    if hasil>kecil then hasil:=kecil;
  end;
  atas := atas+(hasil*range);
  bawah := bawah+hasil;
  Series1.AddXY(range,hasil,'',clTeeColor);
  range := range-8;
  if range>150 then break;
end;
delta := atas/bawah;
end;
'G' : begin
  delta := -75;
end;
'H' : begin
  range:=-300;
  atas:=0;
  bawah:= 0;
  x1:=-300; y1:=0;
  x2:=-150; y2 :=1;
  x3:=-150; y3:=0;
  x4:=0    ; y4 :=1;
  x5:=150  ; y5:=0;

```

Ufuzzy1.pas

```
x6:=0 ; y6 :=1;
Series1.Clear;
while true do
begin
  if (range>=-300) and (range<=-150) then
  begin
    hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);
    if hasil>kecil then hasil:=kecil;
  end
  else if (range>-150) and (range<=0) then
  begin
    hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);
    if hasil>besar then hasil:=besar;
    if hasil<kecil then hasil:=kecil;
  end
  else if (range>0) and (range<=150) then
  begin
    hasil := (y6*(range-x5)-y5*(range-x6))/(x6-x5);
    if hasil>besar then hasil:=besar;
  end;
  atas := atas+(hasil*range);
  bawah := bawah+hasil;
  Series1.AddXY(range,hasil,'',clTeeColor);
  range := range+8;
  if range>150 then break;
end;
delta := atas/bawah;
end;
'I' : begin
  delta := 3.3/1000000000000000000;
end;
'J' : begin
  range:=-150;
  atas:=0;
  bawah:= 0;
  x1:=-150; y1:=0;
  x2:=0 ; y2 :=1;
  x3:=150 ; y3:=0;
  x4:=0 ; y4 :=-1;
  x5:=300 ; y5:=0;
  x6:=150 ; y6 :=1;
  Series1.Clear;
  while true do
  begin
    if (range>=-150) and (range<=0) then
    begin
      hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);
```



```

                                Ufuzzy1.pas
    if hasil>besar then hasil:=besar;
  end
else if (range>0) and (range<=150) then
  begin
    hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);
    if hasil>besar then hasil:=besar;
    if hasil<kecil then hasil:=kecil;
  end
else if (range>150) and (range<=300) then
  begin
    hasil := (y6*(range-x5)-y5*(range-x6))/(x6-x5);
    if hasil>kecil then hasil:=kecil;
  end;
  atas := atas+(hasil*range);
  bawah := bawah+hasil;
  Series1.AddXY(range,hasil,'',clTeeColor);
  range := range+8;
  if range>300 then break;
end;
delta := atas/bawah;
end;
'K' : begin
  delta := 75;
end;
'L' : begin
  range:=-150;
  atas:=0;
  bawah:= 0;
  x1:=-150; y1:=0;
  x2:=0   ; y2 :=1;
  x3:=150 ; y3:=0;
  x4:=0   ; y4 :=1;
  x5:=300 ; y5:=0;
  x6:=150 ; y6 :=1;
  Series1.Clear;
  while true do
  begin
    if (range>=-150) and (range<=0) then
      begin
        hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);
        if hasil>kecil then hasil:=kecil;
      end
    else if (range>0) and (range<=150) then
      begin
        hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);
        if hasil>besar then hasil:=besar;
        if hasil<kecil then hasil:=kecil;
      end
    end
  end
end

```

# Ufuzzy1.pas

```

    end
  else if (range>150) and (range<=300) then
    begin
      hasil := (y6*(range-x5)-y5*(range-x6))/(x6-x5);
      if hasil>besar then hasil:=besar;
    end;
    atas := atas+(hasil*range);
    bawah := bawah+hasil;
    Series1.AddXY(range,hasil,'',clTeeColor);
    range := range+8;
    if range>300 then break;
  end;
  delta := atas/bawah;
end;
'M' : begin
  delta := 150;
end;
'N' : begin
  range:=0;
  atas:=0;
  bawah:= 0;
  x1:=0 ; y1:=0;
  x2:=150 ; y2 :=1;
  x3:=300 ; y3:=0;
  x4:=150 ; y4 :=1;
  Series1.Clear;
  while true do
    begin
      if (range>=0) and (range<=150) then
        begin
          hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);
          if hasil>besar then hasil:=besar;
        end
      else if (range>150) and (range<=300) then
        begin
          hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);
          if hasil>besar then hasil:=besar;
          if hasil<kecil then hasil:=kecil;
        end
      else if (range>300) and (range<=390) then hasil:=kecil;
      atas := atas+(hasil*range);
      bawah := bawah+hasil;
      Series1.AddXY(range,hasil,'',clTeeColor);
      range := range+8;
      if range>390 then break;
    end;
    delta := atas/bawah;
  end;

```

```

end;
'O' : begin
    delta := 215;
end;
'P' : begin
    range:=0;
    atas:=0;
    bawah:= 0;
    x1:=0 ; y1:=0;
    x2:=150 ; y2 :=1;
    x3:=150 ; y3:=0;
    x4:=300 ; y4 :=1;
    Series1.Clear;
    while true do
    begin
        if (range>=0) and (range<=150) then
        begin
            hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);
            if hasil>kecil then hasil:=kecil;
        end
        else if (range>150) and (range<=300) then
        begin
            hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);
            if hasil>besar then hasil:=besar;
            if hasil<kecil then hasil:=kecil;
        end
        else if (range>300) and (range<=390) then hasil:=besar;
        atas := atas+(hasil*range);
        bawah := bawah+hasil;
        Series1.AddXY(range,hasil,'',clTeeColor);
        range := range+9.5;
        if range>390 then break;
    end;
    delta := atas/bawah;
end;
'Q' : begin
    delta := 304;
end;

end;
Writeln(FileR,FormatFloat('0.00E+00',sudut)+' '+FormatFloat('0'
    '+FormatFloat('0.00E+00',delta));
t:=t+1;
sudut := (delta*0.725*exp(-2.21*t)*sin(1.71*t*pi/180))/1.71;
if abs(sudut)<1/100000 then break;
// akhir looping
;
```



Ufuzzy1.pas

```
dit2.Text := FloatToStr(delta);  
CloseFile(FileR);  
emol.Lines.LoadFromFile('c:\joko.txt');
```

```
control sudut rolling
```

```
Ufuzzy2;
```

```
interface
```

```
uses Windows, Messages, SysUtils, Variants, Classes, Graphics, Control  
Forms,  
Dialogs, TeEngine, Series, TeeProcs, Chart, StdCtrls, Buttons, Ex  
ls;
```

```
TSRVst = class(TForm)  
Panel1: TPanel;  
Panel2: TPanel;  
Panel3: TPanel;  
Edit1: TEdit;  
BitBtn1: TBitBtn;  
Chart1: TChart;  
Series1: TFastLineSeries;  
Edit2: TEdit;  
Series2: TFastLineSeries;  
Memo1: TMemo;  
procedure BitBtn1Click(Sender: TObject);  
private  
{ Private declarations }  
public  
{ Public declarations }  
end;
```

```
TSRVst: TFSRVst;
```

```
implementation
```

```
TSRVst.dfm}
```

```
procedure TFSRVst.BitBtn1Click(Sender: TObject);  
var  
sudut, x1, x2, x3, x4, x5, x6, y1, y2, y3, y4, y5, y6, range :  
real;  
Arah : Char;  
besar, kecil : real;  
delta, ttk1, ttk2, hasil, atas, bawah : real;  
t : real;  
FileR : TextFile;  
begin  
if (Abs(StrToFloat(Edit1.Text)) > 45) then  
MessageDlg('Input melebihi range -45>= input <=45....!', mtInform  
ation, [mbOK], 0)
```

# Ufuzzy2.pas

```

n
assignFile(FileR, 'c:\joko.txt');
rewrite(FileR);
writeln(FileR, '=====');
writeln(FileR, '      SR      t      Delta2 ');
writeln(FileR, '=====');
      0.00E+00      1      0.00E+00
writeln(FileR, FormatFloat('0.00E+00', 0) + '      '+FormatFloat('0', t)
      '+FormatFloat('0.00E+00', 0));
Series2.Clear;
=0;
sudut:=0;
Series2.AddXY(t, sudut, '', clTeeColor);
sudut := strtocfloat(Edit1.Text);
=1;
while true do
begin
Series2.AddXY(t, sudut, '', clTeeColor);
// awal looping
ttk1 := 0;
ttk2 := 0;
if (sudut>=-45) and (sudut<=-30) then
begin
daerah := 'A';
end
else if (sudut>-30) and (sudut<-22.5) then
begin
daerah := 'B';
x1:=-30; y1:=0;
x2:=-15; y2:=1;
ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

x3:=-15; y3:=0;
x4:=-30; y4:=1;
ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=-22.5) then
begin
daerah := 'C'
end
else if (sudut>-22.5) and (sudut<-15) then
begin
daerah := 'D';
x1:=-30; y1:=0;
x2:=-15; y2:=1;
ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

x3:=-15; y3:=0;
x4:=-30; y4:=1;
ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
end

```



```

else if (sudut=-15) then
begin
    daerah := 'E'
end
else if (sudut>-15) and (sudut<-7.5) then
begin
    daerah := 'F';
    x1:=-15; y1:=0;
    x2:=0 ; y2:=1;
    ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

    x3:=0 ; y3:=0;
    x4:=-15; y4:=1;
    ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=-7.5) then
begin
    daerah := 'G'
end
else if (sudut>-7.5) and (sudut<0) then
begin
    daerah := 'H';
    x1:=-15; y1:=0;
    x2:=0 ; y2:=1;
    ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

    x3:=0 ; y3:=0;
    x4:=-15; y4:=1;
    ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=0) then
begin
    daerah := 'I'
end
else if (sudut>0) and (sudut<7.5) then
begin
    daerah := 'J';
    x1:=0 ; y1:=0;
    x2:=15 ; y2:=1;
    ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

    x3:=15 ; y3:=0;
    x4:=0 ; y4:=1;
    ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=7.5) then
begin
    daerah := 'K'
end
else if (sudut>7.5) and (sudut<15) then
begin

```

# Ufuzzy2.pas

```

daerah := 'L';
x1:=0 ; y1:=0;
x2:=15 ; y2:=1;
ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

x3:=15 ; y3:=0;
x4:=0 ; y4:=1;
ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=15) then
begin
    daerah := 'M'
end
else if (sudut>15) and (sudut<22.5) then
begin
    daerah := 'N';
x1:=15 ; y1:=0;
x2:=30 ; y2:=1;
ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

x3:=30 ; y3:=0;
x4:=15 ; y4:=1;
ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=22.5) then
begin
    daerah := 'O'
end
else if (sudut>22.5) and (sudut<30) then
begin
    daerah := 'P';
x1:=15 ; y1:=0;
x2:=30 ; y2:=1;
ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

x3:=30 ; y3:=0;
x4:=15 ; y4:=1;
ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut>=30) and (sudut<=45) then
begin
    daerah := 'Q'
end;

if ttk1>ttk2 then
begin
    besar := ttk1;
    kecil := ttk2;
end
else
begin

```

```

    besar := ttk2;
    kecil := ttk1;
end;

case daerah of
'A' : begin
    delta := -224;
    end;
'B' : begin
    range:=-300;
    atas:=0;
    bawah:= 0;
    x1:=-100; y1:=0;
    x2:=-200; y2 :=1;
    x3:=0    ; y3:=0;
    x4:=-100; y4 :=1;
    Series1.Clear;
    while true do
    begin
        if (range>=-300) and (range<=-200) then hasil:=besar
        else if (range>-200) and (range<=-100) then
            begin
                hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);
                if hasil>besar then hasil:=besar;
                if hasil<kecil then hasil:=kecil;
            end
        else if (range>-100) and (range<=0) then
            begin
                hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);
                if hasil>kecil then hasil:=kecil;
            end;
        atas := atas+(hasil*range);
        bawah := bawah+hasil;
        Series1.AddXY(range,hasil,'',clTeeColor);
        range := range+8;
        if range>0 then break;
    end;
    delta := atas/bawah;
    end;
'C' : begin
    delta := -164;
    end;
'D' : begin
    range:=-300;
    atas:=0;
    bawah:= 0;
    x1:=-200; y1:=0;
    x2:=-100; y2 :=1;
    x3:=0    ; y3:=0;
    x4:=-100; y4 :=1;
    Series1.Clear;

```



# Ufuzzy2.pas

```

while true do
begin
  if (range>=-300) and (range<=-200) then hasil:=kecil
  else if (range>-200) and (range<=-100) then
  begin
    hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);
    if hasil>besar then hasil:=besar;
    if hasil<kecil then hasil:=kecil;
  end
  else if (range>-100) and (range<=0) then
  begin
    hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);
    if hasil>besar then hasil:=besar;
  end;
  atas := atas+(hasil*range);
  bawah := bawah+hasil;
  Series1.AddXY(range,hasil,'',clTeeColor);
  range := range+8;
  if range>0 then break;
end;
delta := atas/bawah;
end;
'E' : begin
  delta := -100;
end;
'F' : begin
  range:=-200;
  atas:=0;
  bawah:= 0;
  x1:=-200; y1:=0;
  x2:=-100; y2 :=1;
  x3:=0 ; y3:=0;
  x4:=-100; y4 :=1;
  x5:-100 ; y5:=0;
  x6:=0 ; y6 :=-1;
  Series1.Clear;
  while true do
  begin
    if (range>=-200) and (range<=-100) then
    begin
      hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);
      if hasil>besar then hasil:=besar;
    end
    else if (range>-100) and (range<=0) then
    begin
      hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);
      if hasil>besar then hasil:=besar;
      if hasil<kecil then hasil:=kecil;
    end
    else if (range>0) and (range<=100) then
    begin

```

```

                                Ufuzzy2.pas
        hasil := (y6*(range-x5)-y5*(range-x6))/(x6-x5);
        if hasil>kecil then hasil:=kecil;
    end;
    atas := atas+(hasil*range);
    bawah := bawah+hasil;
    Series1.AddXY(range,hasil,'',clTeeColor);
    range := range-8;
    if range>100 then break;
end;
delta := atas/bawah;
end;
'G' : begin
    delta := -50;
end;
'H' : begin
    range:=-200;
    atas:=0;
    bawah:= 0;
    x1:=-200; y1:=0;
    x2:=-100; y2 :=1;
    x3:=-100; y3:=0;
    x4:=0   ; y4 :=1;
    x5:=100 ; y5:=0;
    x6:=0   ; y6 :=1;
    Series1.Clear;
    while true do
    begin
        if (range>=-200) and (range<=-100) then
        begin
            hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);
            if hasil>kecil then hasil:=kecil;
        end
        else if (range>-100) and (range<=0) then
        begin
            hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);
            if hasil>besar then hasil:=besar;
            if hasil<kecil then hasil:=kecil;
        end
        else if (range>0) and (range<=100) then
        begin
            hasil := (y6*(range-x5)-y5*(range-x6))/(x6-x5);
            if hasil>besar then hasil:=besar;
        end;
        atas := atas+(hasil*range);
        bawah := bawah+hasil;
        Series1.AddXY(range,hasil,'',clTeeColor);
        range := range+8;
        if range>100 then break;
    end;
    delta := atas/bawah;
end;

```

# Ufuzzy2.pas

```

'I' : begin
    delta := -1.17/10000000000000000;
end;
'J' : begin
    range:=-100;
    atas:=0;
    bawah:= 0;
    x1:=-100; y1:=0;
    x2:=0 ; y2 :=1;
    x3:=100 ; y3:=0;
    x4:=0 ; y4 :=1;
    x5:=-200 ; y5:=0;
    x6:=100 ; y6 :=-1;
    Series1.Clear;
    while true do
    begin
        if (range>=-100) and (range<=0) then
        begin
            hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);
            if hasil>besar then hasil:=besar;
        end
        else if (range>0) and (range<=100) then
        begin
            hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);
            if hasil>besar then hasil:=besar;
            if hasil<kecil then hasil:=kecil;
        end
        else if (range>100) and (range<=200) then
        begin
            hasil := (y6*(range-x5)-y5*(range-x6))/(x6-x5);
            if hasil>kecil then hasil:=kecil;
        end;
        atas := atas+(hasil*range);
        bawah := bawah+hasil;
        Series1.AddXY(range,hasil,'',clTeeColor);
        range := range+8;
        if range>200 then break;
    end;
    delta := atas/bawah;
end;
'K' : begin
    delta := 50;
end;
'L' : begin
    range:=-100;
    atas:=0;
    bawah:= 0;
    x1:=-100; y1:=0;
    x2:=0 ; y2 :=1;
    x3:=100 ; y3:=0;
    x4:=0 ; y4 :=1;

```



# Ufuzzy2.pas

```

x5:=200 ; y5:=0;
x6:=100 ; y6 :=1;
  Series1.Clear;
while true do
begin
  if (range>=-100) and (range<=0) then
  begin
    hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);
    if hasil>kecil then hasil:=kecil;
  end
  else if (range>0) and (range<=100) then
  begin
    hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);
    if hasil>besar then hasil:=besar;
    if hasil<kecil then hasil:=kecil;
  end
  else if (range>100) and (range<=200) then
  begin
    hasil := (y6*(range-x5)-y5*(range-x6))/(x6-x5);
    if hasil>besar then hasil:=besar;
  end;
  atas := atas+(hasil*range);
  bawah := bawah+hasil;
  Series1.AddXY(range,hasil,'',clTeeColor);
  range := range+8;
  if range>200 then break;
end;
delta := atas/bawah;
end;
'M' : begin
  delta := 100;
end;
'N' : begin
  range:=0;
  atas:=0;
  bawah:= 0;
  x1:=0 ; y1:=0;
  x2:=100 ; y2 :=1;
  x3:=200 ; y3:=0;
  x4:=100 ; y4 :=1;
  Series1.Clear;
while true do
begin
  if (range>=0) and (range<=100) then
  begin
    hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);
    if hasil>besar then hasil:=besar;
  end
  else if (range>100) and (range<=200) then
  begin
    hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);

```

```

        Ufuzzy2.pas
        if hasil>besar then hasil:=besar;
        if hasil<kecil then hasil:=kecil;
    end
    else if (range>200) and (range<=300) then hasil:=keci

    atas := atas+(hasil*range);
    bawah := bawah+hasil;
    Series1.AddXY(range,hasil,'',clTeeColor);
    range := range+8;
    if range>300 then break;
end;
delta := atas/bawah;
end;
'O' : begin
    delta := 164;
end;
'P' : begin
    range:=0;
    atas:=0;
    bawah:= 0;
    x1:=0 ; y1:=0;
    x2:=100 ; y2 :=1;
    x3:=100 ; y3:=0;
    x4:=200 ; y4 :=1;
    Series1.Clear;
    while true do
    begin
        if (range>=0) and (range<=100) then
        begin
            hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);
            if hasil>kecil then hasil:=kecil;
        end
        else if (range>100) and (range<=200) then
        begin
            hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);
            if hasil>besar then hasil:=besar;
            if hasil<kecil then hasil:=kecil;
        end
        else if (range>200) and (range<=300) then hasil:=besa

        atas := atas+(hasil*range);
        bawah := bawah+hasil;
        Series1.AddXY(range,hasil,'',clTeeColor);
        range := range+9.5;
        if range>300 then break;
    end;
    delta := atas/bawah;
end;
'Q' : begin
    delta := 224;
end;

```

# Ufuzzy2.pas

```
end;  
WriteLn(FileR, FormatFloat('0.00E+00', sudut) + '      ' + FormatFloat(  
t) + '      ' + FormatFloat('0.00E+00', delta));  
t:=t+1;  
sudut := (delta*0.765*exp(-1.09*t)*sin(1.01*t*pi/180))/1.01;  
if abs(sudut)<1/100000 then break;  
// akhir looping  
d;  
it2.Text := FloatToStr(delta);  
CloseFile(FileR);  
mol.Lines.LoadFromFile('c:\joko.txt');
```



```
control sudut pitching
```

```
Ufuzzy3;
```

```
interface
```

```
uses Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms, Dialogs, TeEngine, Series, TeeProcs, Chart, StdCtrls, Buttons, Ex
ports;
```

```
TFSPtvst = class(TForm)
Panel1: TPanel;
Panel2: TPanel;
Panel3: TPanel;
Edit1: TEdit;
BitBtn1: TBitBtn;
Chart1: TChart;
Series1: TFastLineSeries;
Edit2: TEdit;
Series2: TFastLineSeries;
Memo1: TMemo;
procedure BitBtn1Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
```

```
TFSPtvst: TFSPtvst;
```

```
implementation
```

```
($R *.dfm)
```

```
procedure TFSPtvst.BitBtn1Click(Sender: TObject);
var
sudut, x1, x2, x3, x4, x5, x6, y1, y2, y3, y4, y5, y6, range :
```

```
array of Char;
besar, kecil : real;
delta, ttk1, ttk2, hasil, atas, bawah : real;
: real;
fileR : TextFile;
```

```
if (StrToFloat(Edit1.Text)) > 45 then
MessageDlg('Input melebihi range -45>= input <=45....!', mtInform
, [mbOk], 0)
```

```

n
signFile(FileR, 'c:\joko.txt');
write(FileR);
writeln(FileR, '-----');
writeln(FileR, '      SPr      t      Delta3 ');
writeln(FileR, '-----');
          0.00E+00      1      0.00E+00
writeln(FileR, FormatFloat('0.00E+00', 0) + '      '+FormatFloat('0', t)
          '-FormatFloat('0.00E+00', 0));
ries2.Clear;
=0;
dud:=0;
ries2.AddXY(t, sudut, '', clTeeColor);
dud := strtoint(Edit1.Text);
-1;
while true do
begin
Series2.AddXY(t, sudut, '', clTeeColor);
// awal looping
ttk1 := 0;
ttk2 := 0;
if (sudut>=-45) and (sudut<=-30) then
begin
daerah := 'A';
end
else if (sudut>-30) and (sudut<-22.5) then
begin
daerah := 'B';
x1:=-30; y1:=0;
x2:=-15; y2:=1;
ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

x3:=-15; y3:=0;
x4:=-30; y4:=1;
ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=-22.5) then
begin
daerah := 'C'
end
else if (sudut>-22.5) and (sudut<-15) then
begin
daerah := 'D';
x1:=-30; y1:=0;
x2:=-15; y2:=1;
ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

x3:=-15; y3:=0;
x4:=-30; y4:=1;
ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
end

```

```

else if (sudut=-15) then
begin
    daerah := 'E'
end
else if (sudut>-15) and (sudut<-7.5) then
begin
    daerah := 'F';
    x1:=-15; y1:=0;
    x2:=0 ; y2:=1;
    ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

    x3:=0 ; y3:=0;
    x4:=-15; y4:=1;
    ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=-7.5) then
begin
    daerah := 'G'
end
else if (sudut>-7.5) and (sudut<0) then
begin
    daerah := 'H';
    x1:=-15; y1:=0;
    x2:=0 ; y2:=1;
    ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

    x3:=0 ; y3:=0;
    x4:=-15; y4:=1;
    ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=0) then
begin
    daerah := 'I'
end
else if (sudut>0) and (sudut<7.5) then
begin
    daerah := 'J';
    x1:=0 ; y1:=0;
    x2:=15 ; y2:=1;
    ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

    x3:=15 ; y3:=0;
    x4:=0 ; y4:=1;
    ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=7.5) then
begin
    daerah := 'K'
end
else if (sudut>7.5) and (sudut<15) then
begin

```



# Ufuzzy3.pas

```

daerah := 'L';
x1:=0 ; y1:=0;
x2:=15 ; y2:=1;
ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

x3:=15 ; y3:=0;
x4:=0 ; y4:=1;
ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=15) then
begin
    daerah := 'M'
end
else if (sudut>15) and (sudut<22.5) then
begin
    daerah := 'N';
x1:=15 ; y1:=0;
x2:=30 ; y2:=1;
ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

x3:=30 ; y3:=0;
x4:=15 ; y4:=1;
ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=22.5) then
begin
    daerah := 'O'
end
else if (sudut>22.5) and (sudut<30) then
begin
    daerah := 'P';
x1:=15 ; y1:=0;
x2:=30 ; y2:=1;
ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

x3:=30 ; y3:=0;
x4:=15 ; y4:=1;
ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut>=30) and (sudut<=45) then
begin
    daerah := 'Q'
end;

if ttk1>ttk2 then
begin
    besar := ttk1;
    kecil := ttk2;
end
else
begin

```

```

    besar := ttk2;
    kecil := ttk1;
end;

case daerah of
'A' : begin
    delta := -224;
    end;
'B' : begin
    range:=-300;
    atas:=0;
    bawah:= 0;
    x1:=-100; y1:=0;
    x2:=-200; y2 :=1;
    x3:=0    ; y3:=0;
    x4:=-100; y4 :=1;
    Series1.Clear;
    while true do
    begin
        if (range>=-300) and (range<=-200) then hasil:=besar
        else if (range>-200) and (range<=-100) then
            begin
                hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);
                if hasil>besar then hasil:=besar;
                if hasil<kecil then hasil:=kecil;
            end
        else if (range>-100) and (range<=0) then
            begin
                hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);
                if hasil>kecil then hasil:=kecil;
            end;
        atas := atas+(hasil*range);
        bawah := bawah+hasil;
        Series1.AddXY(range,hasil,'',clTeeColor);
        range := range+8;
        if range>0 then break;
    end;
    delta := atas/bawah;
    end;
'C' : begin
    delta := -164;
    end;
'D' : begin
    range:=-300;
    atas:=0;
    bawah:= 0;
    x1:=-200; y1:=0;
    x2:=-100; y2 :=1;
    x3:=0    ; y3:=0;
    x4:=-100; y4 :=1;
    Series1.Clear;

```

# Ufuzzy3.pas

```

while true do
begin
  if (range>=-300) and (range<=-200) then hasil:=kecil
  else if (range>-200) and (range<=-100) then
  begin
    hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);
    if hasil>besar then hasil:=besar;
    if hasil<kecil then hasil:=kecil;
  end
  else if (range>-100) and (range<=0) then
  begin
    hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);
    if hasil>besar then hasil:=besar;
  end;
  atas := atas+(hasil*range);
  bawah := bawah+hasil;
  Series1.AddXY(range,hasil,'',clTeeColor);
  range := range+8;
  if range>0 then break;
end;
delta := atas/bawah;
end;
'E' : begin
  delta := -100;
end;
'F' : begin
  range:=-200;
  atas:=0;
  bawah:= 0;
  x1:=-200; y1:=0;
  x2:=-100; y2 :=1;
  x3:=0 ; y3:=0;
  x4:=-100; y4 :=1;
  x5:=100 ; y5:=0;
  x6:=0 ; y6 :=1;
  Series1.Clear;
  while true do
  begin
    if (range>=-200) and (range<=-100) then
    begin
      hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);
      if hasil>besar then hasil:=besar;
    end
    else if (range>-100) and (range<=0) then
    begin
      hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);
      if hasil>besar then hasil:=besar;
      if hasil<kecil then hasil:=kecil;
    end
    else if (range>0) and (range<=100) then
    begin

```



```

        Ufuzzy3.pas
        hasil := (y6*(range-x5)-y5*(range-x6))/(x6-x5);
        if hasil>kecil then hasil:=kecil;
    end;
    atas := atas+(hasil*range);
    bawah := bawah+hasil;
    Series1.AddXY(range,hasil,'',clTeeColor);
    range := range+8;
    if range>100 then break;
end;
delta := atas/bawah;
end;
'G' : begin
    delta := -50;
end;
'H' : begin
    range:=-200;
    atas:=0;
    bawah:= 0;
    x1:=-200; y1:=0;
    x2:=-100; y2 :=1;
    x3:=-100; y3:=0;
    x4:=0 ; y4 :=1;
    x5:=100 ; y5:=0;
    x6:=0 ; y6 :=1;
    Series1.Clear;
    while true do
    begin
        if (range>=-200) and (range<=-100) then
        begin
            hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);
            if hasil>kecil then hasil:=kecil;
        end
        else if (range>-100) and (range<=0) then
        begin
            hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);
            if hasil>besar then hasil:=besar;
            if hasil<kecil then hasil:=kecil;
        end
        else if (range>0) and (range<=100) then
        begin
            hasil := (y6*(range-x5)-y5*(range-x6))/(x6-x5);
            if hasil>besar then hasil:=besar;
        end;
        atas := atas+(hasil*range);
        bawah := bawah+hasil;
        Series1.AddXY(range,hasil,'',clTeeColor);
        range := range+8;
        if range>100 then break;
    end;
    delta := atas/bawah;
end;

```

```

'I' : begin
    delta := -1.17/10000000000000000;
end;
'J' : begin
    range:=-100;
    atas:=0;
    bawah:= 0;
    x1:=-100; y1:=0;
    x2:=0 ; y2 :=1;
    x3:=100 ; y3:=0;
    x4:=0 ; y4 :=1;
    x5:=-200 ; y5:=0;
    x6:=100 ; y6 :=1;
    Series1.Clear;
    while true do
    begin
        if (range>=-100) and (range<=0) then
        begin
            hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);
            if hasil>besar then hasil:=besar;
        end
        else if (range>0) and (range<=100) then
        begin
            hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);
            if hasil>besar then hasil:=besar;
            if hasil<kecil then hasil:=kecil;
        end
        else if (range>100) and (range<=200) then
        begin
            hasil := (y6*(range-x5)-y5*(range-x6))/(x6-x5);
            if hasil>kecil then hasil:=kecil;
        end;
        atas := atas+(hasil*range);
        bawah := bawah+hasil;
        Series1.AddXY(range,hasil,'',clTeeColor);
        range := range+8;
        if range>200 then break;
    end;
    delta := atas/bawah;
end;
'K' : begin
    delta := 50;
end;
'L' : begin
    range:=-100;
    atas:=0;
    bawah:= 0;
    x1:=-100; y1:=0;
    x2:=0 ; y2 :=1;
    x3:=100 ; y3:=0;
    x4:=0 ; y4 :=1;

```

# Ufuzzy3.pas

```

x5:=200 ; y5:=0;
x6:=100 ; y6 :=1;
Series1.Clear;
while true do
begin
  if (range>=-100) and (range<=0) then
  begin
    hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);
    if hasil>kecil then hasil:=kecil;
  end
  else if (range>0) and (range<=100) then
  begin
    hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);
    if hasil>besar then hasil:=besar;
    if hasil<kecil then hasil:=kecil;
  end
  else if (range>100) and (range<=200) then
  begin
    hasil := (y6*(range-x5)-y5*(range-x6))/(x6-x5);
    if hasil>besar then hasil:=besar;
  end;
  atas := atas+(hasil*range);
  bawah := bawah+hasil;
  Series1.AddXY(range,hasil,'',clTeeColor);
  range := range+8;
  if range>200 then break;
end;
delta := atas/bawah;
end;
'M' : begin
  delta := 100;
end;
'N' : begin
  range:=0;
  atas:=0;
  bawah:= 0;
  x1:=0 ; y1:=0;
  x2:=100 ; y2 :=1;
  x3:=200 ; y3:=0;
  x4:=100 ; y4 :=1;
  Series1.Clear;
  while true do
  begin
    if (range>=0) and (range<=100) then
    begin
      hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);
      if hasil>besar then hasil:=besar;
    end
    else if (range>100) and (range<=200) then
    begin
      hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);

```



```

                                Ufuzzy3.pas
        if hasil>besar then hasil:=besar;
        if hasil<kecil then hasil:=kecil;
    end
    else if (range>200) and (range<=300) then hasil:=keci

        atas := atas+(hasil*range);
        bawah := bawah-hasil;
        Series1.AddXY(range,hasil,'',clTeeColor);
        range := range-8;
        if range>300 then break;
    end;
    delta := atas/bawah;
end;
'O' : begin
    delta := 164;
end;
'P' : begin
    range:=0;
    atas:=0;
    bawah:= 0;
    x1:=0 ; y1:=0;
    x2:=100 ; y2 :=1;
    x3:=100 ; y3:=0;
    x4:=200 ; y4 :=1;
    Series1.Clear;
    while true do
    begin
        if (range>=0) and (range<=100) then
            begin
                hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);
                if hasil>kecil then hasil:=kecil;
            end
        else if (range>100) and (range<=200) then
            begin
                hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);
                if hasil>besar then hasil:=besar;
                if hasil<kecil then hasil:=kecil;
            end
        else if (range>200) and (range<=300) then hasil:=besa

        atas := atas+(hasil*range);
        bawah := bawah+hasil;
        Series1.AddXY(range,hasil,'',clTeeColor);
        range := range-9.5;
        if range>300 then break;
    end;
    delta := atas/bawah;
end;
'Q' : begin
    delta := 224;
end;

```

Ufuzzy3.pas

```
end;
writeln(FileR, FormatFloat('0.00E+00', sudut) + '      ' + FormatFloat(
t) + '      ' + FormatFloat('0.00E+00', delta));
t:=t+1;
sudut := (delta*1.325*exp(-1.74*t)*sin(1.60*t*pi/180))/1.60;
if abs(sudut)<1/100000 then break;
// akhir looping
d;
it2.Text := FloatToStr(delta);
closeFile(FileR);
mol.Lines.LoadFromFile('c:\joko.txt');
```

## LAMPIRAN B :

### Prosedur Run + Listing Program



## PROSEDUR RUN PROGRAM DELPHI 6

Untuk dapat melihat tampilan program delphi untuk control sudut posisi, rolling, dan pitching, serta penafsiran pengaturan bukaan katup, maka langkah-langkah yang harus dilakukan adalah sebagai berikut : (untuk mengetahui berapa tekanan dan lpm-nya dapat dilihat pada gambar 3.1 – 3.2 berdasarkan besar thrust)

1. Buka program delphi, **start – program – delphi 6**.
2. Setelah memasuki program delphi 6, buka **open project** dari menu **file**. Kemudian pilih file dengan nama **“ptafuzzy”** dalam folder ptafuzzy dari directory c:\. Maka akan tampak tampilan awal seperti pada gambar lampiran b.1 seperti berikut setelah dilakukan **run** :

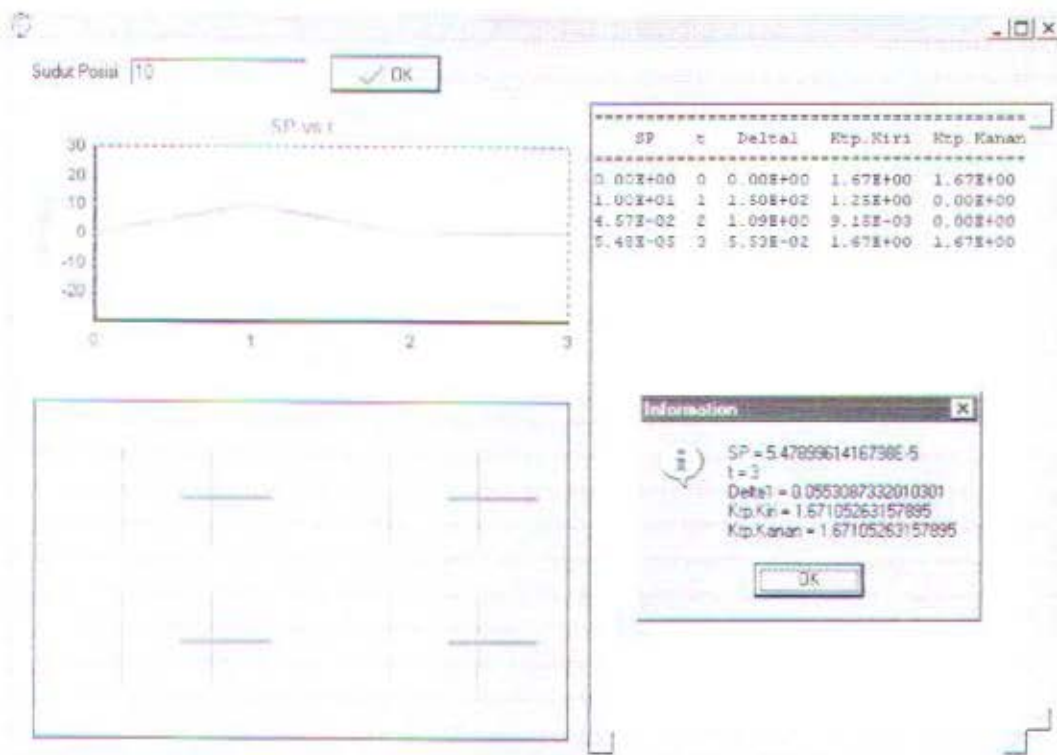


Gb. lampiran b.1a



Gb. lampiran b.1b  
Gb. lampiran b.1 – Tampilan awal

- 3) Dari gambar lampiran b.1b kita dapat melihat tampilan grafik yang kita inginkan dari **menu** (misal grafik SP vs t). Maka akan terlihat tampilan seperti pada gambar lampiran b.2.



Gb. Lampiran b.2 - Grafik hubungan antara SP vs t

Dari gambar lampiran b.2, kita dapat mengetahui posisi bukaan katup seiring perbedaan thrust ( $\Delta I$ ) yang timbul mengacu pada fungsi waktu. Pada detik ke-0,  $\Delta I$  adalah nol dan posisi bukaan katup kiri dan kanan sebesar 1.67 cm dengan thrust tiap-tiap thruster sebesar 200 kgf yang diperlukan untuk gerak maju lurus. Pada detik ke-1,  $\Delta I$  150 kgf yang diperlukan untuk arah belok (sudut posisi) sebesar  $10^0$  dengan posisi bukaan katup kiri sebesar 1.25 cm dan posisi bukaan katup kanan sebesar 0 cm. Begitu seterusnya sampai pada detik ke-3 dengan  $\Delta I$  sebesar  $5.53 \times 10^{-2}$  kgf dan sudut posisi sebesar  $5.48 \times 10^{-5}$  deg ( $S_p = 0^0$  jika  $S_p \leq 10^{-3}$ ) dan posisi bukaan katup sudah kembali pada posisi semula dengan besar bukaan 1.67 cm untuk tiap-tiap katup, dan berarti rov sudah berjalan lurus pada arah belok  $10^0$  dari posisi semula.

Begitu halnya jika kita melihat pada grafik hubungan antara SR vs t atau pada grafik hubungan antara  $S_{Pt}$  vs t dari **menu** yang diperlihatkan pada gambar lampiran b.2, kita akan mengetahui posisi bukaan katup yang terjadi untuk mengatasi masukan yang ada agar diperoleh hasil yang diinginkan.



```

Ufuzzy1.pas
/ control sudut posisi + bukaan katup

unit Ufuzzy1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Co
  ntrols, Forms,
  Dialogs, TeEngine, Series, TeeProcs, Chart, StdCtrls, Button
  ExtCtrls;

type
TFSPvst = class(TForm)
  Panel1: TPanel;
  Panel2: TPanel;
  Panel3: TPanel;
  Edit1: TEdit;
  BitBtn1: TBitBtn;
  Edit2: TEdit;
  Memo1: TMemo;
  Panel4: TPanel;
  Chart1: TChart;
  Series1: TFastLineSeries;
  Series2: TFastLineSeries;
  Chart2: TChart;
  Series3: TFastLineSeries;
  Series4: TFastLineSeries;
  Series5: TFastLineSeries;
  Series6: TFastLineSeries;
  Series7: TFastLineSeries;
  Series8: TFastLineSeries;
  Series9: TFastLineSeries;
  Series10: TFastLineSeries;
  Series11: TFastLineSeries;
  Series12: TFastLineSeries;
  Series13: TFastLineSeries;
  Series14: TFastLineSeries;
  procedure BitBtn1Click(Sender: TObject);
  procedure FormShow(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

FSPvst: TFSPvst;

implementation

```

```
SR *.dfm);
```

```

procedure TFSPvst.BitBtn1Click(Sender: TObject);
var sudut, x1, x2, x3, x4, x5, x6, y1, y2, y3, y4, y5, y6, ran
a : real;
    daerah : Char;
    besar, kecil : real;
    delta, ttk1, ttk2, hasil, atas, bawah : real;
    t : real;
    FileR : TextFile;
    z1, z2 : real;
    delay1, delay2 : integer;
begin
    if (abs(StrToFloat(Edit1.Text))>30) then
        MessageDlg('Input melebihi range -30>= input <=30....!', mtI
formation, [mbOk], 0)
    else
        begin
            Memol.Text := '';
            AssignFile(FileR, 'c:\joko.txt');
            Rewrite(FileR);
            Writeln(FileR, '=====');
            Writeln(FileR, '      SP      t      Delta1      Ktp.Kiri      Ktp.Kanan');
            Writeln(FileR, '=====');
            Writeln(FileR, '0.00E+00  1  0.00E+00  0.00E+00  0.00E+00');
            Memol.Lines.Add('=====');

            Memol.Lines.Add('      SP      t      Delta1      Ktp.Kiri      Ktp.Kanan'

            Memol.Lines.Add('=====');

            Series2.Clear;
            t:=0;
            sudut:=0;
            Series2.AddXY(t, sudut, '', clTeeColor);
            Series3.Clear;
            Series4.Clear;
            Series5.Clear;
            Series6.Clear;
            Series7.Clear;
            Series8.Clear;
            Series9.Clear;
            Series10.Clear;
            Series11.Clear;
            Series12.Clear;
            Series13.Clear;
            Series14.Clear;
            Series3.AddXY(10, 10, '', clTeeColor);
            Series3.AddXY(10, 30, '', clTeeColor);
            Series4.AddXY(20, 10, '', clTeeColor);
            Series4.AddXY(20, 30, '', clTeeColor);

```

# Ufuzzy1.pas

```

z1:=200*10/304;
Series11.AddXY(10+z1,20,'',clTeeColor);
Series11.AddXY(10+z1+10,20,'',clTeeColor);

Series5.AddXY(40,10,'',clTeeColor);
Series5.AddXY(40,30,'',clTeeColor);
Series6.AddXY(50,10,'',clTeeColor);
Series6.AddXY(50,30,'',clTeeColor);
z2:=200*10/304;
Series12.AddXY(40+z2,20,'',clTeeColor);
Series12.AddXY(40+z2+10,20,'',clTeeColor);

Series7.AddXY(10,40,'',clTeeColor);
Series7.AddXY(10,60,'',clTeeColor);
Series8.AddXY(20,40,'',clTeeColor);
Series8.AddXY(20,60,'',clTeeColor);
Series13.AddXY(10+z1,50,'',clTeeColor);
Series13.AddXY(10+z1+10,50,'',clTeeColor);

Series9.AddXY(40,40,'',clTeeColor);
Series9.AddXY(40,60,'',clTeeColor);
Series10.AddXY(50,40,'',clTeeColor);
Series10.AddXY(50,60,'',clTeeColor);
Series14.AddXY(40+z2,50,'',clTeeColor);
Series14.AddXY(40+z2+10,50,'',clTeeColor);
Writeln(FileR,FormatFloat('0.00E+00',0)+' '+FormatFloat('0'
)+'+FormatFloat('0.00E+00',0)+' '+FormatFloat('0.00E+00'
1*0.254)+' '-FormatFloat('0.00E+00',z2*0.254));
Memo1.Lines.Add(FormatFloat('0.00E+00',0)+' '+FormatFloat('
,t)+' '+FormatFloat('0.00E+00',0)+' '+FormatFloat('0.00E+0
,z1*0.254)+' '+FormatFloat('0.00E+00',z2*0.254));
sudut := strtofloat(Edit1.Text);
t:=1;
while true do
begin
  Series2.AddXY(t,sudut,'',clTeeColor);
  // awal looping
  ttk1 := 0;
  ttk2 := 0;
  if (sudut>=-30) and (sudut<=-20) then
  begin
    daerah := 'A';
  end
  else if (sudut>-20) and (sudut<=-15) then
  begin
    daerah := 'B';
    x1:=-20; y1:=0;
    x2:=-10; y2:=1;
    ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);
    x3:=-10; y3:=0;

```



Ufuzzy1.pas

```
x4:=-20; y4:=1;
ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=-15) then
begin
daerah := 'C'
end
else if (sudut>-15) and (sudut<-10) then
begin
daerah := 'D';
x1:=-20; y1:=0;
x2:=-10; y2:=1;
ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

x3:=-10; y3:=0;
x4:=-20; y4:=1;
ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=-10) then
begin
daerah := 'E'
end
else if (sudut>-10) and (sudut<-5) then
begin
daerah := 'F';
x1:=-10; y1:=0;
x2:=0 ; y2:=1;
ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

x3:=0 ; y3:=0;
x4:=-10; y4:=1;
ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=-5) then
begin
daerah := 'G'
end
else if (sudut>-5) and (sudut<0) then
begin
daerah := 'H';
x1:=-10; y1:=0;
x2:=0 ; y2:=1;
ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

x3:=0 ; y3:=0;
x4:=-10; y4:=1;
ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=0) then
begin
daerah := 'I'
```

```

end
else if (sudut>0) and (sudut<5) then
begin
    daerah := 'J';
    x1:=0 ; y1:=0;
    x2:=10 ; y2:=1;
    ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

    x3:=10 ; y3:=0;
    x4:=0 ; y4:=1;
    ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=5) then
begin
    daerah := 'K'
end
else if (sudut>5) and (sudut<10) then
begin
    daerah := 'L';
    x1:=0 ; y1:=0;
    x2:=10 ; y2:=1;
    ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

    x3:=10 ; y3:=0;
    x4:=0 ; y4:=1;
    ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=10) then
begin
    daerah := 'M'
end
else if (sudut>10) and (sudut<15) then
begin
    daerah := 'N';
    x1:=10 ; y1:=0;
    x2:=20 ; y2:=1;
    ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

    x3:=20 ; y3:=0;
    x4:=10 ; y4:=1;
    ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=15) then
begin
    daerah := 'O'
end
else if (sudut>15) and (sudut<20) then
begin
    daerah := 'P';
    x1:=10 ; y1:=0;
    x2:=20 ; y2:=1;

```

```

                                Ufuzzy1.pas
ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

x3:=20 ; y3:=0;
x4:=10 ; y4:=1;
ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut>=20) and (sudut<=30) then
begin
    daerah := 'Q'
end;

if ttk1>ttk2 then
begin
    besar := ttk1;
    kecil := ttk2;
end
else
begin
    besar := ttk2;
    kecil := ttk1;
end;

case daerah of
'A' : begin
    delta := -304;
    end;
'B' : begin
    range:=-390;
    atas:=0;
    bawah:= 0;
    x1:=-150; y1:=0;
    x2:=-300; y2 :=1;
    x3:=0 ; y3:=0;
    x4:=-150; y4 :=1;
    Series1.Clear;
    while true do
    begin
        if (range>=-390) and (range<=-300) then hasil:=b
        else if (range>-300) and (range<=-150) then
            begin
                hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x
                if hasil>besar then hasil:=besar;
                if hasil<kecil then hasil:=kecil;
            end
        else if (range>-150) and (range<=0) then
            begin
                hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x
                if hasil>kecil then hasil:=kecil;

```



# Ufuzzy1.pas

```

        end;
        atas := atas+(hasil*range);
        bawah := bawah+hasil;
        Series1.AddXY(range,hasil,'',clTeeColor);
        range := range+8;
        if range>0 then break;
    end;
    delta := atas/bawah;
end;
'C' : begin
    delta := -215;
end;
'D' : begin
    range:=-390;
    atas:=0;
    bawah:= 0;
    x1:=-300; y1:=0;
    x2:=-150; y2 :=1;
    x3:=0    ; y3:=0;
    x4:=-150; y4 :=1;
    Series1.Clear;
    while true do
    begin
        if (range>=-390) and (range<=-300) then hasil:=k
        else if (range>-300) and (range<=-150) then
        begin
            hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1)
            if hasil>besar then hasil:=besar;
            if hasil<kecil then hasil:=kecil;
        end
        else if (range>-150) and (range<=0) then
        begin
            hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3)
            if hasil>besar then hasil:=besar;
        end;
        atas := atas+(hasil*range);
        bawah := bawah+hasil;
        Series1.AddXY(range,hasil,'',clTeeColor);
        range := range+8;
        if range>0 then break;
    end;
    delta := atas/bawah;
end;
'E' : begin
    delta := -150;
end;
'F' : begin
    range:=-300;

```

# Ufuzzyl.pas

```

atas:=0;
bawah:= 0;
x1:=-300; y1:=0;
x2:=-150; y2 :=1;
x3:=0    ; y3:=0;
x4:=-150; y4 :=1;
x5:=150  ; y5:=0;
x6:=0    ; y6 :=1;
Series1.Clear;
while true do
begin
  if (range>=-300) and (range<=-150) then
  begin
    hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);

    if hasil>besar then hasil:=besar;
  end
  else if (range>-150) and (range<=0) then
  begin
    hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);

    if hasil>besar then hasil:=besar;
    if hasil<kecil then hasil:=kecil;
  end
  else if (range>0) and (range<=150) then
  begin
    hasil := (y6*(range-x5)-y5*(range-x6))/(x6-x5);

    if hasil>kecil then hasil:=kecil;
  end;
  atas := atas+(hasil*range);
  bawah := bawah+hasil;
  Series1.AddXY(range,hasil,'',clTeeColor);
  range := range+8;
  if range>150 then break;
end;
delta := atas/bawah;
end;
'G' : begin
  delta := -75;
end;
'H' : begin
  range:=-300;
  atas:=0;
  bawah:= 0;
  x1:=-300; y1:=0;
  x2:=-150; y2 :=1;
  x3:=-150; y3:=0;
  x4:=0    ; y4 :=1;
  x5:=150  ; y5:=0;
  x6:=0    ; y6 :=1;

```

# Ufuzzyl.pas

```

Series1.Clear;
while true do
begin
    if (range>=-300) and (range<=-150) then
    begin
        hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);

        if hasil>kecil then hasil:=kecil;
    end
    else if (range>-150) and (range<=0) then
    begin
        hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);

        if hasil>besar then hasil:=besar;
        if hasil<kecil then hasil:=kecil;
    end
    else if (range>0) and (range<=150) then
    begin
        hasil := (y6*(range-x5)-y5*(range-x6))/(x6-x5);

        if hasil>besar then hasil:=besar;
    end;
    atas := atas+(hasil*range);
    bawah := bawah+hasil;
    Series1.AddXY(range,hasil,'',clTeeColor);
    range := range+8;
    if range>150 then break;
end;
delta := atas/bawah;
end;
'I' : begin
    delta := 3.3/10000000000000000;
end;
'J' : begin
    range:=-150;
    atas:=0;
    bawah:= 0;
    x1:=-150; y1:=0;
    x2:=0 ; y2 :=1;
    x3:=150 ; y3:=0;
    x4:=0 ; y4 :=1;
    x5:=300 ; y5:=0;
    x6:=150 ; y6 :=1;
    Series1.Clear;
    while true do
    begin
        if (range>=-150) and (range<=0) then
        begin
            hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);

            if hasil>besar then hasil:=besar;

```



# Ufuzzyl.pas

```

end
else if (range>0) and (range<=150) then
begin
    hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);

    if hasil>besar then hasil:=besar;
    if hasil<kecil then hasil:=kecil;
end
else if (range>150) and (range<=300) then
begin
    hasil := (y6*(range-x5)-y5*(range-x6))/(x6-x5);

    if hasil>kecil then hasil:=kecil;
end;
atas := atas+(hasil*range);
bawah := bawah+hasil;
Series1.AddXY(range,hasil,'',clTeeColor);
range := range+8;
if range>300 then break;
end;
delta := atas/bawah;
end;
'K' : begin
    delta := 75;
end;
'L' : begin
    range:=-150;
    atas:=0;
    bawah:= 0;
    x1:=-150; y1:=0;
    x2:=0 ; y2 :=1;
    x3:=150 ; y3:=0;
    x4:=0 ; y4 :=1;
    x5:=300 ; y5:=0;
    x6:=150 ; y6 :=1;
    Series1.Clear;
    while true do
    begin
        if (range>=-150) and (range<=0) then
        begin
            hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);

            if hasil>kecil then hasil:=kecil;
        end
        else if (range>0) and (range<=150) then
        begin
            hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);

            if hasil>besar then hasil:=besar;
            if hasil<kecil then hasil:=kecil;
        end
    end
end

```

```

                                Ufuzzy1.pas
else if (range>150) and (range<=300) then
begin
    hasil := (y6*(range-x5)-y5*(range-x6))/(x6-x5);

    if hasil>besar then hasil:=besar;
end;
atas := atas+(hasil*range);
bawah := bawah+hasil;
Series1.AddXY(range,hasil,'',clTeeColor);
range := range+8;
if range>300 then break;
end;
delta := atas/bawah;
end;
'M' : begin
    delta := 150;
end;
'N' : begin
    range:=0;
    atas:=0;
    bawah:= 0;
    x1:=0 ; y1:=0;
    x2:=150 ; y2 :=1;
    x3:=300 ; y3:=0;
    x4:=150 ; y4 :=1;
    Series1.Clear;
    while true do
    begin
        if (range>=0) and (range<=150) then
        begin
            hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);

            if hasil>besar then hasil:=besar;
        end
        else if (range>150) and (range<=300) then
        begin
            hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);

            if hasil>besar then hasil:=besar;
            if hasil<kecil then hasil:=kecil;
        end
        else if (range>300) and (range<=390) then hasil:=
            kecil;

            atas := atas+(hasil*range);
            bawah := bawah+hasil;
            Series1.AddXY(range,hasil,'',clTeeColor);
            range := range+8;
            if range>390 then break;
        end;
        delta := atas/bawah;
    end;
end;

```

```

'O' : begin
    delta := 215;
end;
'P' : begin
    range:=0;
    atas:=0;
    bawah:= 0;
    x1:=0 ; y1:=0;
    x2:=150 ; y2 :=1;
    x3:=150 ; y3:=0;
    x4:=300 ; y4 :=1;
    Series1.Clear;
    while true do
    begin
        if (range>=0) and (range<=150) then
        begin
            hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);

            if hasil>kecil then hasil:=kecil;
        end
        else if (range>150) and (range<=300) then
        begin
            hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);

            if hasil>besar then hasil:=besar;
            if hasil<kecil then hasil:=kecil;
        end
        else if (range>300) and (range<=390) then hasil:=
esar;

        atas := atas+(hasil*range);
        bawah := bawah+hasil;
        Series1.AddXY(range,hasil,'',clTeeColor);
        range := range+9.5;
        if range>390 then break;
    end;
    delta := atas/bawah;
end;
'Q' : begin
    delta := 304;
end;

end;
Series11.Clear;
Series12.Clear;
Series13.Clear;
Series14.Clear;

if delta>0 then
begin
    if abs(delta)<=0.1 then z1:=200*10/304
    else z1:=abs(delta)*10/304;

```



```

                                Ufuzzy1.pas
Series11.AddXY(10+z1,20,'',clTeeColor);
Series11.AddXY(10+z1+10,20,'',clTeeColor);
Series13.AddXY(10+z1,50,'',clTeeColor);
Series13.AddXY(10+z1+10,50,'',clTeeColor);

if abs(delta)<=0.1 then z2:=200*10/304
else z2:=0;
Series12.AddXY(40+z2,20,'',clTeeColor);
Series12.AddXY(40+z2+10,20,'',clTeeColor);
Series14.AddXY(40+z2,50,'',clTeeColor);
Series14.AddXY(40+z2+10,50,'',clTeeColor);
end
else
begin
if abs(delta)<=0.1 then z1:=200*10/304
else z1:=0;
Series11.AddXY(10+z1,20,'',clTeeColor);
Series11.AddXY(10+z1+10,20,'',clTeeColor);
Series13.AddXY(10+z1,50,'',clTeeColor);
Series13.AddXY(10+z1+10,50,'',clTeeColor);

if abs(delta)<=0.1 then z2:=200*10/304
else z2:=abs(delta)*10/304;
Series12.AddXY(40+z2,20,'',clTeeColor);
Series12.AddXY(40+z2+10,20,'',clTeeColor);
Series14.AddXY(40+z2,50,'',clTeeColor);
Series14.AddXY(40+z2+10,50,'',clTeeColor);
end;
WriteLn(FileR,FormatFloat('0.00E+00',sudut)+' '+FormatFlo
('0',t)+' '+FormatFloat('0.00E+00',delta)+' '+FormatFloat(
'0.00E+00',z1*0.254)+' '+FormatFloat('0.00E+00',z2*0.254));
Memo1.Lines.Add(FormatFloat('0.00E+00',sudut)+' '+FormatF
at('0',t)+' '+FormatFloat('0.00E+00',delta)+' '+FormatFloa
t('0.00E+00',z1*0.254)+' '+FormatFloat('0.00E+00',z2*0.254));
MessageDlgPos('SP = '+floattostr(sudut)+chr(10)+chr(13)+'t
'+FormatFloat('0',t)+chr(10)+chr(13)+'Delta1 = '+floattostr
(delta)+chr(10)+chr(13)+'Ktp.Kiri = '+floattostr(z1*0.254)+chr
(10)+chr(13)+'Ktp.Kanan = '+floattostr(z2*0.254), mtInformatio
[mbOk], 0, 470, 300);
t:=t+1;
sudut := (delta*0.725*exp(-2.21*t)*sin(1.71*t*pi/180))/1.7

if (abs(sudut)<1/100000) and (abs(delta)<1/10) then break;
// akhir looping
end;
Edit2.Text := FloatToStr(delta);
CloseFile(FileR);
Memo1.Lines.LoadFromFile('c:\joko.txt');
;
;

```

# Ufuzzy1.pas

```

procedure TFSPvst.FormShow(Sender: TObject);
var z1, z2 : real;
begin
  Memo1.Text:='';
  Series3.Clear;
  Series4.Clear;
  Series5.Clear;
  Series6.Clear;
  Series7.Clear;
  Series8.Clear;
  Series9.Clear;
  Series10.Clear;
  Series11.Clear;
  Series12.Clear;
  Series13.Clear;
  Series14.Clear;
  Series3.AddXY(10,10,'',clTeeColor);
  Series3.AddXY(10,30,'',clTeeColor);
  Series4.AddXY(20,10,'',clTeeColor);
  Series4.AddXY(20,30,'',clTeeColor);
  z1:=200*10/304;
  Series11.AddXY(10+z1,20,'',clTeeColor);
  Series11.AddXY(10+z1+10,20,'',clTeeColor);

  Series5.AddXY(40,10,'',clTeeColor);
  Series5.AddXY(40,30,'',clTeeColor);
  Series6.AddXY(50,10,'',clTeeColor);
  Series6.AddXY(50,30,'',clTeeColor);
  z2:=200*10/304;
  Series12.AddXY(40+z2,20,'',clTeeColor);
  Series12.AddXY(40+z2+10,20,'',clTeeColor);

  Series7.AddXY(10,40,'',clTeeColor);
  Series7.AddXY(10,60,'',clTeeColor);
  Series8.AddXY(20,40,'',clTeeColor);
  Series8.AddXY(20,60,'',clTeeColor);
  Series13.AddXY(10+z1,50,'',clTeeColor);
  Series13.AddXY(10+z1+10,50,'',clTeeColor);

  Series9.AddXY(40,40,'',clTeeColor);
  Series9.AddXY(40,60,'',clTeeColor);
  Series10.AddXY(50,40,'',clTeeColor);
  Series10.AddXY(50,60,'',clTeeColor);
  Series14.AddXY(40+z2,50,'',clTeeColor);
  Series14.AddXY(40+z2+10,50,'',clTeeColor);
end;

```

```

                                Ufuzzy2.pas
/ control sudut rolling + bukaan katup

unit Ufuzzy2;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Co
  ntrols, Forms,
  Dialogs, TeEngine, Series, TeeProcs, Chart, StdCtrls, Button
  ExtCtrls;

type
  TFSRvst = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Panel3: TPanel;
    Edit1: TEdit;
    BitBtn1: TBitBtn;
    Edit2: TEdit;
    Memo1: TMemo;
    Panel4: TPanel;
    Chart1: TChart;
    Series1: TFastLineSeries;
    Series2: TFastLineSeries;
    Chart2: TChart;
    Series3: TFastLineSeries;
    Series4: TFastLineSeries;
    Series5: TFastLineSeries;
    Series6: TFastLineSeries;
    Series7: TFastLineSeries;
    Series8: TFastLineSeries;
    Series9: TFastLineSeries;
    Series10: TFastLineSeries;
    Series11: TFastLineSeries;
    Series12: TFastLineSeries;
    Series13: TFastLineSeries;
    Series14: TFastLineSeries;
    procedure BitBtn1Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FSRvst: TFSRvst;

implementation

```



```
SR *.dfm}
```

```

procedure TFSRvst.BitBtn1Click(Sender: TObject);
var sudut, x1, x2, x3, x4, x5, x6, y1, y2, y3, y4, y5, y6, ran
e : real;
    daerah : Char;
    besar, kecil : real;
    delta, ttk1, ttk2, hasil, atas, bawah : real;
    t : real;
    FileR : TextFile;
    z1, z2 : real;
    delay1, delay2 : integer;
begin
    if (abs(StrToFloat(Edit1.Text)) > 45) then
        MessageDlg('Input melebihi range -45 >= input <= 45....!', mtI
formation, [mbOk], 0)
    else
        begin
            Mem01.Text := '';
            AssignFile(FileR, 'c:\joko.txt');
            Rewrite(FileR);
            Writeln(FileR, '=====');
            Writeln(FileR, '      SR      t      Delta2      Ktp.Kiri      Ktp.Kanan');
            Writeln(FileR, '=====');
            Writeln(FileR, '0.00E+00  1  0.00E+00  0.00E+00  0.00E+00');
            Mem01.Lines.Add('=====');

            Mem01.Lines.Add('      SR      t      Delta2      Ktp.Kiri      Ktp.Kanan'

            Mem01.Lines.Add('=====');

            Series2.Clear;
            t := 0;
            sudut := 0;
            Series2.AddXY(t, sudut, '', clTeeColor);
            Series3.Clear;
            Series4.Clear;
            Series5.Clear;
            Series6.Clear;
            Series7.Clear;
            Series8.Clear;
            Series9.Clear;
            Series10.Clear;
            Series11.Clear;
            Series12.Clear;
            Series13.Clear;
            Series14.Clear;
            Series3.AddXY(10, 10, '', clTeeColor);
            Series3.AddXY(10, 30, '', clTeeColor);
            Series4.AddXY(20, 10, '', clTeeColor);
            Series4.AddXY(20, 30, '', clTeeColor);

```

# Ufuzzy2.pas

```

z1:=0*10/224;
Series11.AddXY(10+z1,20,'',clTeeColor);
Series11.AddXY(10+z1+10,20,'',clTeeColor);

Series5.AddXY(40,10,'',clTeeColor);
Series5.AddXY(40,30,'',clTeeColor);
Series6.AddXY(50,10,'',clTeeColor);
Series6.AddXY(50,30,'',clTeeColor);
z2:=0*10/224;
Series12.AddXY(40+z2,20,'',clTeeColor);
Series12.AddXY(40+z2+10,20,'',clTeeColor);

Series7.AddXY(10,40,'',clTeeColor);
Series7.AddXY(10,60,'',clTeeColor);
Series8.AddXY(20,40,'',clTeeColor);
Series8.AddXY(20,60,'',clTeeColor);
Series13.AddXY(10+z1,50,'',clTeeColor);
Series13.AddXY(10+z1+10,50,'',clTeeColor);

Series9.AddXY(40,40,'',clTeeColor);
Series9.AddXY(40,60,'',clTeeColor);
Series10.AddXY(50,40,'',clTeeColor);
Series10.AddXY(50,60,'',clTeeColor);
Series14.AddXY(40+z2,50,'',clTeeColor);
Series14.AddXY(40+z2+10,50,'',clTeeColor);
Writeln(FileR,FormatFloat('0.00E+00',0)+' '+FormatFloat('0'
)+'+FormatFloat('0.00E+00',0)+' '+FormatFloat('0.00E+00'
1*0.254)+' '+FormatFloat('0.00E+00',z2*0.254));
Memol.Lines.Add(FormatFloat('0.00E+00',0)+' '+FormatFloat('
,t)+' '+FormatFloat('0.00E+00',0)+' '+FormatFloat('0.00E+0
,z1*0.254)+' '+FormatFloat('0.00E+00',z2*0.254));
sudut := strtofloat(Edit1.Text);
t:=1;
while true do
begin
    Series2.AddXY(t,sudut,'',clTeeColor);
    // awal looping
    ttk1 := 0;
    ttk2 := 0;
    if (sudut>=-45) and (sudut<=-30) then
    begin
        daerah := 'A';
    end
    else if (sudut>-30) and (sudut<-22.5) then
    begin
        daerah := 'B';
        x1:=-30; y1:=0;
        x2:=-15; y2:=1;
        ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

        x3:=-15; y3:=0;
    end
end

```

# Ufuzzy2.pas

```

    x4:=-30; y4:=1;
    ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=-22.5) then
begin
    daerah := 'C'
end
else if (sudut>-22.5) and (sudut<-15) then
begin
    daerah := 'D';
    x1:=-30; y1:=0;
    x2:=-15; y2:=1;
    ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

    x3:=-15; y3:=0;
    x4:=-30; y4:=1;
    ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=-15) then
begin
    daerah := 'E'
end
else if (sudut>-15) and (sudut<-7.5) then
begin
    daerah := 'F';
    x1:=-15; y1:=0;
    x2:=0 ; y2:=1;
    ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

    x3:=0 ; y3:=0;
    x4:=-15; y4:=1;
    ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=-7.5) then
begin
    daerah := 'G'
end
else if (sudut>-7.5) and (sudut<0) then
begin
    daerah := 'H';
    x1:=-15; y1:=0;
    x2:=0 ; y2:=1;
    ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

    x3:=0 ; y3:=0;
    x4:=-15; y4:=1;
    ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=0) then
begin
    daerah := 'I'

```



```

end
else if (sudut>0) and (sudut<7.5) then
begin
    daerah := 'J';
    x1:=0 ; y1:=0;
    x2:=15 ; y2:=1;
    ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

    x3:=15 ; y3:=0;
    x4:=0 ; y4:=1;
    ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=7.5) then
begin
    daerah := 'K'
end
else if (sudut>7.5) and (sudut<15) then
begin
    daerah := 'L';
    x1:=0 ; y1:=0;
    x2:=15 ; y2:=1;
    ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

    x3:=15 ; y3:=0;
    x4:=0 ; y4:=1;
    ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=15) then
begin
    daerah := 'M'
end
else if (sudut>15) and (sudut<22.5) then
begin
    daerah := 'N';
    x1:=15 ; y1:=0;
    x2:=30 ; y2:=1;
    ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

    x3:=30 ; y3:=0;
    x4:=15 ; y4:=1;
    ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=22.5) then
begin
    daerah := 'O'
end
else if (sudut>22.5) and (sudut<30) then
begin
    daerah := 'P';
    x1:=15 ; y1:=0;
    x2:=30 ; y2:=1;

```

```

                                Ufuzzy2.pas
ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

x3:=30 ; y3:=0;
x4:=15 ; y4:=1;
ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut>=30) and (sudut<=45) then
begin
    daerah := 'Q'
end;

if ttk1>ttk2 then
begin
    besar := ttk1;
    kecil := ttk2;
end
else
begin
    besar := ttk2;
    kecil := ttk1;
end;

case daerah of
'A' : begin
    delta := -224;
    end;
'B' : begin
    range:=-300;
    atas:=0;
    bawah:= 0;
    x1:=-100; y1:=0;
    x2:=-200; y2 :=1;
    x3:=0 ; y3:=0;
    x4:=-100; y4 :=1;
    Series1.Clear;
    while true do
    begin
        if (range>=-300) and (range<=-200) then hasil:=b
        else if (range>-200) and (range<=-100) then
            begin
                hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x
                if hasil>besar then hasil:=besar;
                if hasil<kecil then hasil:=kecil;
            end
        else if (range>-100) and (range<=0) then
            begin
                hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x
                if hasil>kecil then hasil:=kecil;

```

# Ufuzzy2.pas

```

        end;
        atas := atas+(hasil*range);
        bawah := bawah+hasil;
        Series1.AddXY(range,hasil,'',clTeeColor);
        range := range+8;
        if range>0 then break;
    end;
    delta := atas/bawah;
end;
'C' : begin
    delta := -164;
end;
'D' : begin
    range:=-300;
    atas:=0;
    bawah:= 0;
    x1:=-200; y1:=0;
    x2:=-100; y2 :=1;
    x3:=0    ; y3:=0;
    x4:=-100; y4 :=-1;
    Series1.Clear;
    while true do
    begin
        if (range>=-300) and (range<=-200) then hasil:=k
        else if (range>-200) and (range<=-100) then
        begin
            hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1)
            if hasil>besar then hasil:=besar;
            if hasil<kecil then hasil:=kecil;
        end
        else if (range>-100) and (range<=0) then
        begin
            hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3)
            if hasil>besar then hasil:=besar;
        end;
        atas := atas+(hasil*range);
        bawah := bawah+hasil;
        Series1.AddXY(range,hasil,'',clTeeColor);
        range := range+8;
        if range>0 then break;
    end;
    delta := atas/bawah;
end;
'E' : begin
    delta := -100;
end;
'F' : begin
    range:=-200;

```



# Ufuzzy2.pas

```

atas:=0;
bawah:= 0;
x1:=-200; y1:=0;
x2:=-100; y2 :=1;
x3:=0 ; y3:=0;
x4:=-100; y4 :=1;
x5:=100 ; y5:=0;
x6:=0 ; y6 :=1;
Series1.Clear;
while true do
begin
  if (range>=-200) and (range<=-100) then
  begin
    hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);

    if hasil>besar then hasil:=besar;
  end
  else if (range>-100) and (range<=0) then
  begin
    hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);

    if hasil>besar then hasil:=besar;
    if hasil<kecil then hasil:=kecil;
  end
  else if (range>0) and (range<=100) then
  begin
    hasil := (y6*(range-x5)-y5*(range-x6))/(x6-x5);

    if hasil>kecil then hasil:=kecil;
  end;
  atas := atas+(hasil*range);
  bawah := bawah+hasil;
  Series1.AddXY(range,hasil,'',clTeeColor);
  range := range+8;
  if range>100 then break;
end;
delta := atas/bawah;
end;
'G' : begin
  delta := -50;
end;
'H' : begin
  range:=-200;
  atas:=0;
  bawah:= 0;
  x1:=-200; y1:=0;
  x2:=-100; y2 :=1;
  x3:=-100; y3:=0;
  x4:=0 ; y4 :=1;
  x5:=100 ; y5:=0;
  x6:=0 ; y6 :=1;

```

# Ufuzzy2.pas

```

Series1.Clear;
while true do
begin
    if (range>=-200) and (range<=-100) then
    begin
        hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);

        if hasil>kecil then hasil:=kecil;
    end
    else if (range>-100) and (range<=0) then
    begin
        hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);

        if hasil>besar then hasil:=besar;
        if hasil<kecil then hasil:=kecil;
    end
    else if (range>0) and (range<=100) then
    begin
        hasil := (y6*(range-x5)-y5*(range-x6))/(x6-x5);

        if hasil>besar then hasil:=besar;
    end;
    atas := atas+(hasil*range);
    bawah := bawah+hasil;
    Series1.AddXY(range,hasil,'',clTeeColor);
    range := range+8;
    if range>100 then break;
end;
delta := atas/bawah;
end;
'I' : begin
    delta := -1.17/10000000000000000;
end;
'J' : begin
    range:=-100;
    atas:=0;
    bawah:= 0;
    x1:=-100; y1:=0;
    x2:=0 ; y2 :=1;
    x3:=100 ; y3:=0;
    x4:=0 ; y4 :=1;
    x5:=200 ; y5:=0;
    x6:=100 ; y6 :=1;
    Series1.Clear;
    while true do
    begin
        if (range>=-100) and (range<=0) then
        begin
            hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);

            if hasil>besar then hasil:=besar;

```

# Ufuzzy2.pas

```

end
else if (range>0) and (range<=100) then
begin
    hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);

    if hasil>besar then hasil:=besar;
    if hasil<kecil then hasil:=kecil;
end
else if (range>100) and (range<=200) then
begin
    hasil := (y6*(range-x5)-y5*(range-x6))/(x6-x5);

    if hasil>kecil then hasil:=kecil;
end;
atas := atas+(hasil*range);
bawah := bawah+hasil;
Series1.AddXY(range,hasil,'',clTeeColor);
range := range+8;
if range>200 then break;
end;
delta := atas/bawah;
end;
'K' : begin
    delta := 50;
end;
'L' : begin
    range:=-100;
    atas:=0;
    bawah:= 0;
    x1:=-100; y1:=0;
    x2:=0 ; y2 :=-1;
    x3:=100 ; y3:=0;
    x4:=0 ; y4 :=1;
    x5:=200 ; y5:=0;
    x6:=100 ; y6 :=1;
    Series1.Clear;
    while true do
    begin
        if (range>=-100) and (range<=0) then
        begin
            hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);

            if hasil>kecil then hasil:=kecil;
        end
        else if (range>0) and (range<=100) then
        begin
            hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);

            if hasil>besar then hasil:=besar;
            if hasil<kecil then hasil:=kecil;
        end
    end
end

```



```

                                Ufuzzy2.pas
else if (range>100) and (range<=200) then
begin
    hasil := (y6*(range-x5)-y5*(range-x6))/(x6-x5);

    if hasil>besar then hasil:=besar;
end;
atas := atas+(hasil*range);
bawah := bawah+hasil;
    Series1.AddXY(range,hasil,'',clTeeColor);
range := range+8;
if range>200 then break;
end;
delta := atas/bawah;
end;
'M' : begin
    delta := 100;
end;
'N' : begin
    range:=0;
    atas:=0;
    bawah:= 0;
    x1:=0 ; y1:=0;
    x2:=100 ; y2 :=1;
    x3:=200 ; y3:=0;
    x4:=100 ; y4 :=1;
    Series1.Clear;
    while true do
    begin
        if (range>=0) and (range<=100) then
        begin
            hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);

            if hasil>besar then hasil:=besar;
        end
        else if (range>100) and (range<=200) then
        begin
            hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);

            if hasil>besar then hasil:=besar;
            if hasil<kecil then hasil:=kecil;
        end
        else if (range>200) and (range<=300) then hasil:=
kecil;

        atas := atas+(hasil*range);
        bawah := bawah+hasil;
        Series1.AddXY(range,hasil,'',clTeeColor);
        range := range+8;
        if range>300 then break;
    end;
    delta := atas/bawah;
end;

```

```

'O' : begin
    delta := 164;
end;
'P' : begin
    range:=0;
    atas:=0;
    bawah:= 0;
    x1:=0 ; y1:=0;
    x2:=100 ; y2 :=1;
    x3:=100 ; y3:=0;
    x4:=200 ; y4 :=1;
    Series1.Clear;
    while true do
    begin
        if (range>=0) and (range<=100) then
        begin
            hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);

            if hasil>kecil then hasil:=kecil;
        end
        else if (range>100) and (range<=200) then
        begin
            hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);

            if hasil>besar then hasil:=besar;
            if hasil<kecil then hasil:=kecil;
        end
        else if (range>200) and (range<=300) then hasil:=
esar;

        atas := atas+(hasil*range);
        bawah := bawah+hasil;
        Series1.AddXY(range,hasil,'',clTeeColor);
        range := range+9.5;
        if range>300 then break;
    end;
    delta := atas/bawah;
end;
'Q' : begin
    delta := 224;
end;

```

```

end;
Series11.Clear;
Series12.Clear;
Series13.Clear;
Series14.Clear;

if delta<0 then
begin
    if abs(delta)<0.01 then z1:=0
    else z1:=abs(delta)*10/224;

```

# Ufuzzy2.pas

```

Series11.AddXY(10+z1,20,'',clTeeColor);
Series11.AddXY(10+z1+10,20,'',clTeeColor);
Series13.AddXY(10+z1,50,'',clTeeColor);
Series13.AddXY(10+z1+10,50,'',clTeeColor);

z2:=0;
Series12.AddXY(40+z2,20,'',clTeeColor);
Series12.AddXY(40+z2+10,20,'',clTeeColor);
Series14.AddXY(40+z2,50,'',clTeeColor);
Series14.AddXY(40+z2+10,50,'',clTeeColor);
end
else
begin
  z1:=0;
  Series11.AddXY(10+z1,20,'',clTeeColor);
  Series11.AddXY(10+z1+10,20,'',clTeeColor);
  Series13.AddXY(10+z1,50,'',clTeeColor);
  Series13.AddXY(10+z1+10,50,'',clTeeColor);

  if abs(delta)<0.01 then z2:=0
  else z2:=abs(delta)*10/224;
  Series12.AddXY(40+z2,20,'',clTeeColor);
  Series12.AddXY(40+z2+10,20,'',clTeeColor);
  Series14.AddXY(40+z2,50,'',clTeeColor);
  Series14.AddXY(40+z2+10,50,'',clTeeColor);
end;
  Writeln(FileR,FormatFloat('0.00E+00',sudut)+' '+FormatFlo
('0',t)+' '+FormatFloat('0.00E+00',delta)+' '+FormatFloat(
.00E+00',z1*0.254)+' '+FormatFloat('0.00E+00',z2*0.254));
  Memol.Lines.Add(FormatFloat('0.00E+00',sudut)+' '+FormatF
at('0',t)+' '+FormatFloat('0.00E+00',delta)+' '+FormatFloa
'0.00E+00',z1*0.254)+' '+FormatFloat('0.00E+00',z2*0.254));
  MessageDlgPos('SR = '+floattostr(sudut)+chr(10)+chr(13)+'t
'+FormatFloat('0',t)+chr(10)+chr(13)+'Delta1 = '+floattostr
elta)+chr(10)+chr(13)+'Ktp.Kiri = '+floattostr(z1*0.254)+chr
0)+chr(13)+'Ktp.Kanan = '+floattostr(z2*0.254), mtInformatio
[mbOk], 0, 470, 300);
  t:=t+1;
  sudut := (delta*0.765*exp(-1.09*t)*sin(1.01*t*pi/180))/1.0

  if abs(sudut)<1/100000 then break;
  // akhir looping
end;
Edit2.Text := FloatToStr(delta);
CloseFile(FileR);
Memol.Lines.LoadFromFile('c:\joko.txt');
#;
#;

cedure TFSRvst.FormShow(Sender: TObject);

```



```

var z1,z2 : real;
begin
  Series3.Clear;
  Series4.Clear;
  Series5.Clear;
  Series6.Clear;
  Series7.Clear;
  Series8.Clear;
  Series9.Clear;
  Series10.Clear;
  Series11.Clear;
  Series12.Clear;
  Series13.Clear;
  Series14.Clear;
  Series3.AddXY(10,10,'',clTeeColor);
  Series3.AddXY(10,30,'',clTeeColor);
  Series4.AddXY(20,10,'',clTeeColor);
  Series4.AddXY(20,30,'',clTeeColor);
  z1:=0*10/224;
  Series11.AddXY(10+z1,20,'',clTeeColor);
  Series11.AddXY(10+z1+10,20,'',clTeeColor);

  Series5.AddXY(40,10,'',clTeeColor);
  Series5.AddXY(40,30,'',clTeeColor);
  Series6.AddXY(50,10,'',clTeeColor);
  Series6.AddXY(50,30,'',clTeeColor);
  z2:=0*10/224;
  Series12.AddXY(40+z2,20,'',clTeeColor);
  Series12.AddXY(40+z2+10,20,'',clTeeColor);

  Series7.AddXY(10,40,'',clTeeColor);
  Series7.AddXY(10,60,'',clTeeColor);
  Series8.AddXY(20,40,'',clTeeColor);
  Series8.AddXY(20,60,'',clTeeColor);
  Series13.AddXY(10+z1,50,'',clTeeColor);
  Series13.AddXY(10+z1+10,50,'',clTeeColor);

  Series9.AddXY(40,40,'',clTeeColor);
  Series9.AddXY(40,60,'',clTeeColor);
  Series10.AddXY(50,40,'',clTeeColor);
  Series10.AddXY(50,60,'',clTeeColor);
  Series14.AddXY(40+z2,50,'',clTeeColor);
  Series14.AddXY(40+z2+10,50,'',clTeeColor);
end;

```

```

                                Ufuzzy3.pas
/ control sudut pitching + bukaan katup

unit Ufuzzy3;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Co
  ntrols, Forms,
  Dialogs, TeEngine, Series, TeeProcs, Chart, StdCtrls, Button
  ExtCtrls;

type
  TFSptvst = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Panel3: TPanel;
    Edit1: TEdit;
    BitBtn1: TBitBtn;
    Edit2: TEdit;
    Memo1: TMemo;
    Panel4: TPanel;
    Chart1: TChart;
    Series1: TFastLineSeries;
    Series2: TFastLineSeries;
    Chart2: TChart;
    Series3: TFastLineSeries;
    Series4: TFastLineSeries;
    Series5: TFastLineSeries;
    Series6: TFastLineSeries;
    Series7: TFastLineSeries;
    Series8: TFastLineSeries;
    Series9: TFastLineSeries;
    Series10: TFastLineSeries;
    Series11: TFastLineSeries;
    Series12: TFastLineSeries;
    Series13: TFastLineSeries;
    Series14: TFastLineSeries;
    procedure BitBtn1Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FSptvst: TFSptvst;

implementation

```



```
SR *.dfm)
```

```

procedure TFSPTvst.BitBtn1Click(Sender: TObject);
var sudut, x1, x2, x3, x4, x5, x6, y1, y2, y3, y4, y5, y6, ran
e : real;
    daerah : Char;
    besar, kecil : real;
    delta, ttk1, ttk2, hasil, atas, bawah : real;
    t : real;
    FileR : TextFile;
    z1, z2 : real;
    delay1, delay2 : integer;
begin
    if (abs(StrToFloat(Edit1.Text))>45) then
        MessageDlg('Input melebihi range -45>= input <=45.....!', mtI
formation, [mbOk], 0)
    else
        begin
            Memol.Text := '';
            AssignFile(FileR, 'c:\joko.txt');
            Rewrite(FileR);
            Writeln(FileR, '=====');
            Writeln(FileR, '      Spt      t      Delta3      Ktp.Depan      Ktp.Blk ');
            Writeln(FileR, '=====');
            Writeln(FileR, '0.00E+00      1      0.00E+00      0.00E+00      0.00E+00');
            Memol.Lines.Add('=====');

            Memol.Lines.Add('      Spt      t      Delta3      Ktp.Depan      Ktp.Blk ');

            Memol.Lines.Add('=====');

            Series2.Clear;
            t:=0;
            sudut:=0;
            Series2.AddXY(t, sudut, '', clTeeColor);
            Series3.Clear;
            Series4.Clear;
            Series5.Clear;
            Series6.Clear;
            Series7.Clear;
            Series8.Clear;
            Series9.Clear;
            Series10.Clear;
            Series11.Clear;
            Series12.Clear;
            Series13.Clear;
            Series14.Clear;
            Series3.AddXY(25, 10, '', clTeeColor);
            Series3.AddXY(25, 30, '', clTeeColor);
            Series4.AddXY(35, 10, '', clTeeColor);
            Series4.AddXY(35, 30, '', clTeeColor);

```



# Ufuzzy3.pas

```

z1:=0*10/224;
Series11.AddXY(25+z1,20,'',clTeeColor);
Series11.AddXY(25+z1+10,20,'',clTeeColor);

Series5.AddXY(40,10,'',clTeeColor);
Series5.AddXY(40,30,'',clTeeColor);
Series6.AddXY(50,10,'',clTeeColor);
Series6.AddXY(50,30,'',clTeeColor);
z2:=0*10/224;
Series12.AddXY(40+z2,20,'',clTeeColor);
Series12.AddXY(40+z2+10,20,'',clTeeColor);

Series7.AddXY(10,40,'',clTeeColor);
Series7.AddXY(10,60,'',clTeeColor);
Series8.AddXY(20,40,'',clTeeColor);
Series8.AddXY(20,60,'',clTeeColor);
Series13.AddXY(10+z1,50,'',clTeeColor);
Series13.AddXY(10+z1+10,50,'',clTeeColor);

Series9.AddXY(40,40,'',clTeeColor);
Series9.AddXY(40,60,'',clTeeColor);
Series10.AddXY(50,40,'',clTeeColor);
Series10.AddXY(50,60,'',clTeeColor);
Series14.AddXY(40+z2,50,'',clTeeColor);
Series14.AddXY(40+z2+10,50,'',clTeeColor);
Writeln(FileR,FormatFloat('0.00E+00',0)+' '+FormatFloat('0'
)+'+FormatFloat('0.00E+00',0)+' '+FormatFloat('0.00E+00'
1*0.254)+' '+FormatFloat('0.00E+00',z2*0.254));
Memor1.Lines.Add(FormatFloat('0.00E+00',0)+' '+FormatFloat('
,t)+' '+FormatFloat('0.00E+00',0)+' '+FormatFloat('0.00E+0
,z1*0.254)+' '+FormatFloat('0.00E+00',z2*0.254));
sudut := strtofloat(Edit1.Text);
t:=1;
while true do
begin
  Series2.AddXY(t,sudut,'',clTeeColor);
  // awal looping
  ttk1 := 0;
  ttk2 := 0;
  if (sudut>=-45) and (sudut<=-30) then
  begin
    daerah := 'A';
  end
  else if (sudut>-30) and (sudut<-22.5) then
  begin
    daerah := 'B';
    x1:=-30; y1:=0;
    x2:=-15; y2:=1;
    ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

    x3:=-15; y3:=0;
  end
end

```

# Ufuzzy3.pas

```

    x4:=-30; y4:=1;
    ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=-22.5) then
begin
    daerah := 'C'
end
else if (sudut>-22.5) and (sudut<-15) then
begin
    daerah := 'D';
    x1:=-30; y1:=0;
    x2:=-15; y2:=1;
    ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

    x3:=-15; y3:=0;
    x4:=-30; y4:=1;
    ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=-15) then
begin
    daerah := 'E'
end
else if (sudut>-15) and (sudut<-7.5) then
begin
    daerah := 'F';
    x1:=-15; y1:=0;
    x2:=0 ; y2:=1;
    ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

    x3:=0 ; y3:=0;
    x4:=-15; y4:=1;
    ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=-7.5) then
begin
    daerah := 'G'
end
else if (sudut>-7.5) and (sudut<0) then
begin
    daerah := 'H';
    x1:=-15; y1:=0;
    x2:=0 ; y2:=1;
    ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

    x3:=0 ; y3:=0;
    x4:=-15; y4:=1;
    ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=0) then
begin
    daerah := 'I'

```

# Ufuzzy3.pas

```

end
else if (sudut>0) and (sudut<7.5) then
begin
    daerah := 'J';
    x1:=0 ; y1:=0;
    x2:=15 ; y2:=1;
    ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

    x3:=15 ; y3:=0;
    x4:=0 ; y4:=1;
    ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=7.5) then
begin
    daerah := 'K'
end
else if (sudut>7.5) and (sudut<15) then
begin
    daerah := 'L';
    x1:=0 ; y1:=0;
    x2:=15 ; y2:=1;
    ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

    x3:=15 ; y3:=0;
    x4:=0 ; y4:=1;
    ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=15) then
begin
    daerah := 'M'
end
else if (sudut>15) and (sudut<22.5) then
begin
    daerah := 'N';
    x1:=15 ; y1:=0;
    x2:=30 ; y2:=1;
    ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

    x3:=30 ; y3:=0;
    x4:=15 ; y4:=1;
    ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut=22.5) then
begin
    daerah := 'O'
end
else if (sudut>22.5) and (sudut<30) then
begin
    daerah := 'P';
    x1:=15 ; y1:=0;
    x2:=30 ; y2:=1;

```



```

                                Ufuzzy3.pas
ttk1:=(y2*(sudut-x1)-y1*(sudut-x2))/(x2-x1);

x3:=30 ; y3:=0;
x4:=15 ; y4:=1;
ttk2:=(y4*(sudut-x3)-y3*(sudut-x4))/(x4-x3);
end
else if (sudut>=30) and (sudut<=45) then
begin
    daerah := 'Q'
end;

if ttk1>ttk2 then
begin
    besar := ttk1;
    kecil := ttk2;
end
else
begin
    besar := ttk2;
    kecil := ttk1;
end;

case daerah of
'A' : begin
    delta := -224;
end;
'B' : begin
    range:=-300;
    atas:=0;
    bawah:= 0;
    x1:=-100; y1:=0;
    x2:=-200; y2 :=1;
    x3:=0 ; y3:=0;
    x4:=-100; y4 :=1;
    Series1.Clear;
    while true do
    begin
        if (range>=-300) and (range<=-200) then hasil:=b
        else if (range>-200) and (range<=-100) then
            begin
                hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x
                if hasil>besar then hasil:=besar;
                if hasil<kecil then hasil:=kecil;
            end
        else if (range>-100) and (range<=0) then
            begin
                hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x
                if hasil>kecil then hasil:=kecil;

```

# Ufuzzy3.pas

```

        end;
        atas := atas+(hasil*range);
        bawah := bawah+hasil;
        Series1.AddXY(range,hasil,'',clTeeColor);
        range := range-8;
        if range>0 then break;
    end;
    delta := atas/bawah;
end;
'C' : begin
    delta := -164;
end;
'D' : begin
    range:=-300;
    atas:=0;
    bawah:= 0;
    x1:=-200; y1:=0;
    x2:=-100; y2 :=1;
    x3:=0    ; y3:=0;
    x4:=-100; y4 :=1;
    Series1.Clear;
    while true do
    begin
        if (range>=-300) and (range<=-200) then hasil:=k

        else if (range>-200) and (range<=-100) then
        begin
            hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1)

            if hasil>besar then hasil:=besar;
            if hasil<kecil then hasil:=kecil;
        end
        else if (range>-100) and (range<=0) then
        begin
            hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3)

            if hasil>besar then hasil:=besar;
        end;
        atas := atas+(hasil*range);
        bawah := bawah+hasil;
        Series1.AddXY(range,hasil,'',clTeeColor);
        range := range+8;
        if range>0 then break;
    end;
    delta := atas/bawah;
end;
'E' : begin
    delta := -100;
end;
'F' : begin
    range:=-200;

```

# Ufuzzy3.pas

```

atas:=0;
bawah:= 0;
x1:=-200; y1:=0;
x2:=-100; y2 :=1;
x3:=0 ; y3:=0;
x4:=-100; y4 :=1;
x5:=-100 ; y5:=0;
x6:=0 ; y6 :=1;
Series1.Clear;
while true do
begin
  if (range>=-200) and (range<=-100) then
  begin
    hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);

    if hasil>besar then hasil:=besar;
  end
  else if (range>-100) and (range<=0) then
  begin
    hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);

    if hasil>besar then hasil:=besar;
    if hasil<kecil then hasil:=kecil;
  end
  else if (range>0) and (range<=100) then
  begin
    hasil := (y6*(range-x5)-y5*(range-x6))/(x6-x5);

    if hasil>kecil then hasil:=kecil;
  end;
  atas := atas+(hasil*range);
  bawah := bawah+hasil;
  Series1.AddXY(range,hasil,'',clTeeColor);
  range := range+8;
  if range>100 then break;
end;
delta := atas/bawah;
end;
'G' : begin
  delta := -50;
end;
'H' : begin
  range:=-200;
  atas:=0;
  bawah:= 0;
  x1:=-200; y1:=0;
  x2:=-100; y2 :=1;
  x3:=-100; y3:=0;
  x4:=0 ; y4 :=1;
  x5:=100 ; y5:=0;
  x6:=0 ; y6 :=1;

```



# Ufuzzy3.pas

```

Series1.Clear;
while true do
begin
  if (range>=-200) and (range<=-100) then
  begin
    hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);

    if hasil>kecil then hasil:=kecil;
  end
  else if (range>-100) and (range<=0) then
  begin
    hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);

    if hasil>besar then hasil:=besar;
    if hasil<kecil then hasil:=kecil;
  end
  else if (range>0) and (range<=100) then
  begin
    hasil := (y6*(range-x5)-y5*(range-x6))/(x6-x5);

    if hasil>besar then hasil:=besar;
  end;
  atas := atas+(hasil*range);
  bawah := bawah+hasil;
  Series1.AddXY(range,hasil,'',clTeeColor);
  range := range+8;
  if range>100 then break;
end;
delta := atas/bawah;
end;
'I' : begin
  delta := -1.17/10000000000000000;
end;
'J' : begin
  range:=-100;
  atas:=0;
  bawah:= 0;
  x1:=-100; y1:=0;
  x2:=0 ; y2 :=1;
  x3:=100 ; y3:=0;
  x4:=0 ; y4 :=1;
  x5:=200 ; y5:=0;
  x6:=100 ; y6 :=1;
  Series1.Clear;
  while true do
  begin
    if (range>=-100) and (range<=0) then
    begin
      hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);

      if hasil>besar then hasil:=besar;

```

# Ufuzzy3.pas

```

end
else if (range>0) and (range<=100) then
begin
    hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);

    if hasil>besar then hasil:=besar;
    if hasil<kecil then hasil:=kecil;
end
else if (range>100) and (range<=200) then
begin
    hasil := (y6*(range-x5)-y5*(range-x6))/(x6-x5);

    if hasil>kecil then hasil:=kecil;
end;
atas := atas+(hasil*range);
bawah := bawah+hasil;
Series1.AddXY(range,hasil,'',clTeeColor);
range := range+8;
if range>200 then break;
end;
delta := atas/bawah;
end;
'K' : begin
    delta := 50;
end;
'L' : begin
    range:=-100;
    atas:=0;
    bawah:= 0;
    x1:=-100; y1:=0;
    x2:=0 ; y2 :=1;
    x3:=100 ; y3:=0;
    x4:=0 ; y4 :=-1;
    x5:=200 ; y5:=0;
    x6:=100 ; y6 :=1;
    Series1.Clear;
    while true do
    begin
        if (range>=-100) and (range<=0) then
        begin
            hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);

            if hasil>kecil then hasil:=kecil;
        end
        else if (range>0) and (range<=100) then
        begin
            hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);

            if hasil>besar then hasil:=besar;
            if hasil<kecil then hasil:=kecil;
        end
    end
end

```

```

                                Ufuzzy3.pas
else if (range>100) and (range<=200) then
begin
    hasil := (y6*(range-x5)-y5*(range-x6))/(x6-x5);

    if hasil>besar then hasil:=besar;
end;
atas := atas+(hasil*range);
bawah := bawah+hasil;
    Series1.AddXY(range,hasil,'',clTeeColor);
range := range+8;
if range>200 then break;
end;
delta := atas/bawah;
end;
'M' : begin
    delta := 100;
end;
'N' : begin
    range:=0;
    atas:=0;
    bawah:= 0;
    x1:=0 ; y1:=0;
    x2:=100 ; y2 :=1;
    x3:=200 ; y3:=0;
    x4:=100 ; y4 :=1;
    Series1.Clear;
    while true do
    begin
        if (range>=0) and (range<=100) then
        begin
            hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);

            if hasil>besar then hasil:=besar;
        end
        else if (range>100) and (range<=200) then
        begin
            hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);

            if hasil>besar then hasil:=besar;
            if hasil<kecil then hasil:=kecil;
        end
        else if (range>200) and (range<=300) then hasil:
            := (y6*(range-x5)-y5*(range-x6))/(x6-x5);

            if hasil>besar then hasil:=besar;
            if hasil<kecil then hasil:=kecil;
        end
        atas := atas+(hasil*range);
        bawah := bawah+hasil;
        Series1.AddXY(range,hasil,'',clTeeColor);
        range := range+8;
        if range>300 then break;
    end;
    delta := atas/bawah;
end;

```



```

'O' : begin
    delta := 164;
end;
'P' : begin
    range:=0;
    atas:=0;
    bawah:= 0;
    x1:=0 ; y1:=0;
    x2:=100 ; y2 :=1;
    x3:=100 ; y3:=0;
    x4:=200 ; y4 :=1;
    Series1.Clear;
    while true do
    begin
        if (range>=0) and (range<=100) then
        begin
            hasil := (y2*(range-x1)-y1*(range-x2))/(x2-x1);

            if hasil>kecil then hasil:=kecil;
        end
        else if (range>100) and (range<=200) then
        begin
            hasil := (y4*(range-x3)-y3*(range-x4))/(x4-x3);

            if hasil>besar then hasil:=besar;
            if hasil<kecil then hasil:=kecil;
        end
        else if (range>200) and (range<=300) then hasil:=
esar;

        atas := atas+(hasil*range);
        bawah := bawah+hasil;
        Series1.AddXY(range,hasil,'',clTeeColor);
        range := range+9.5;
        if range>300 then break;
    end;
    delta := atas/bawah;
end;
'Q' : begin
    delta := 224;
end;

end;
Series11.Clear;
Series12.Clear;
Series13.Clear;
Series14.Clear;

if delta<0 then
begin
    if abs(delta)<0.01 then z1:=0
    else z1:=abs(delta)*10/224;

```

# Ufuzzy3.pas

```

Series11.AddXY(25+z1,20,'',clTeeColor);
Series11.AddXY(25+z1+10,20,'',clTeeColor);

z2:=0;
Series12.AddXY(40+z2,20,'',clTeeColor);
Series12.AddXY(40+z2+10,20,'',clTeeColor);
Series13.AddXY(10+z2,50,'',clTeeColor);
Series13.AddXY(10+z2+10,50,'',clTeeColor);
Series14.AddXY(40+z2,50,'',clTeeColor);
Series14.AddXY(40+z2+10,50,'',clTeeColor);
end
else
begin
  z1:=0;
  Series11.AddXY(25+z1,20,'',clTeeColor);
  Series11.AddXY(25+z1+10,20,'',clTeeColor);

  if abs(delta)<0.01 then z2:=0
  else z2:=abs(delta)*10/224;
  Series12.AddXY(40+z2,20,'',clTeeColor);
  Series12.AddXY(40+z2+10,20,'',clTeeColor);
  Series13.AddXY(10+z2,50,'',clTeeColor);
  Series13.AddXY(10+z2+10,50,'',clTeeColor);
  Series14.AddXY(40+z2,50,'',clTeeColor);
  Series14.AddXY(40+z2+10,50,'',clTeeColor);
end;
Writeln(FileR,FormatFloat('0.00E+00',sudut)+' '+FormatFlo
('0',t)+' '+FormatFloat('0.00E+00',delta)+' '+FormatFloat(
.00E+00',z1*0.254)+' '+FormatFloat('0.00E+00',z2*0.254));
Memol.Lines.Add(FormatFloat('0.00E+00',sudut)+' '+FormatF
at('0',t)+' '+FormatFloat('0.00E+00',delta)+' '+FormatFlea
'0.00E+00',z1*0.254)+' '+FormatFloat('0.00E+00',z2*0.254));
MessageDlgPos('SPt = '+floattostr(sudut)+chr(10)+chr(13)+'
'+FormatFloat('0',t)+chr(10)+chr(13)+'Deltal = '+floattost
delta)+chr(10)+chr(13)+'Ktp.Kiri = '+floattostr(z1*0.254)+ch
10)+chr(13)+'Ktp.Kanan = '+floattostr(z2*0.254), mtInformati
[mbOk], 0, 470, 300);
t:=t+1;
sudut := (delta*1.325*exp(-1.74*t)*sin(1.60*t*pi/180))/1.6

if abs(sudut)<1/100000 then break;
// akhir looping
end;
Edit2.Text := FloatToStr(delta);
CloseFile(FileR);
Memol.Lines.LoadFromFile('c:\joko.txt');
;
;

procedure TFSptvst.FormShow(Sender: TObject);

```

```

var z1,z2 : real;
begin
  Series3.Clear;
  Series4.Clear;
  Series5.Clear;
  Series6.Clear;
  Series7.Clear;
  Series8.Clear;
  Series9.Clear;
  Series10.Clear;
  Series11.Clear;
  Series12.Clear;
  Series13.Clear;
  Series14.Clear;
  Series3.AddXY(25,10,'',clTeeColor);
  Series3.AddXY(25,30,'',clTeeColor);
  Series4.AddXY(35,10,'',clTeeColor);
  Series4.AddXY(35,30,'',clTeeColor);
  z1:=0*10/224;
  Series11.AddXY(25+z1,20,'',clTeeColor);
  Series11.AddXY(25+z1+10,20,'',clTeeColor);

  Series5.AddXY(40,10,'',clTeeColor);
  Series5.AddXY(40,30,'',clTeeColor);
  Series6.AddXY(50,10,'',clTeeColor);
  Series6.AddXY(50,30,'',clTeeColor);
  z2:=0*10/224;
  Series12.AddXY(40+z2,20,'',clTeeColor);
  Series12.AddXY(40+z2+10,20,'',clTeeColor);

  Series7.AddXY(10,40,'',clTeeColor);
  Series7.AddXY(10,60,'',clTeeColor);
  Series8.AddXY(20,40,'',clTeeColor);
  Series8.AddXY(20,60,'',clTeeColor);
  Series13.AddXY(10+z1,50,'',clTeeColor);
  Series13.AddXY(10+z1+10,50,'',clTeeColor);

  Series9.AddXY(40,40,'',clTeeColor);
  Series9.AddXY(40,60,'',clTeeColor);
  Series10.AddXY(50,40,'',clTeeColor);
  Series10.AddXY(50,60,'',clTeeColor);
  Series14.AddXY(40+z2,50,'',clTeeColor);
  Series14.AddXY(40+z2+10,50,'',clTeeColor);
end;

```



# LAMPIRAN C :

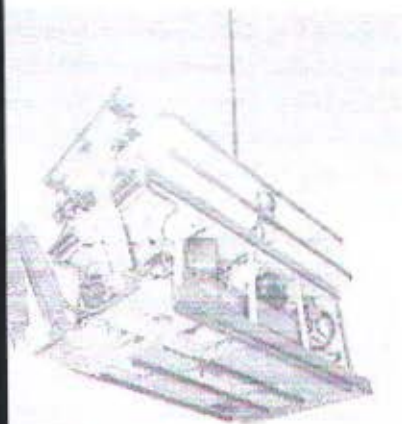
## Data ROV

## Products and Services

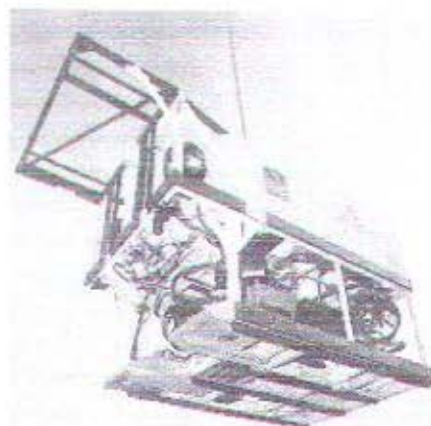
### Underwater Remotely Operated Vehicles (ROVs)

Hydrovision's core business is the design and manufacture of ROVs, trenching and cable maintenance systems and subsea robotics.

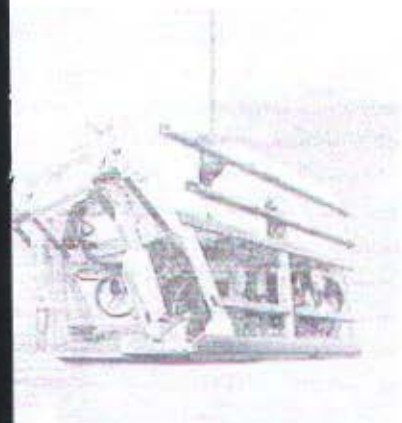
Standard Hydrovision ROV products include the Demon, Diablo & Venom work class ROVs and Venom T cable maintenance and burial systems.



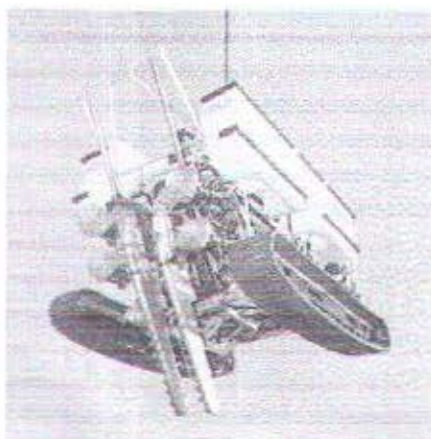
Diablo - 100 shp, 2000 msw



Demon - 100 shp, 3000 msw



Venom 100 or 150 shp

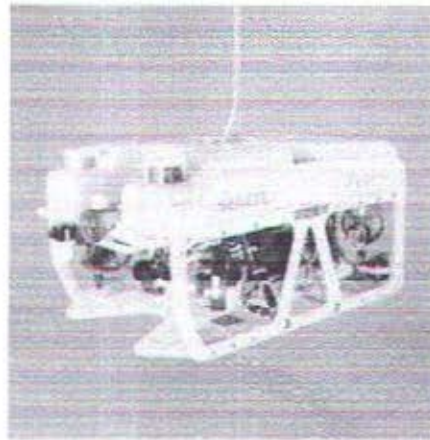


Venom 3K-450T cable maintenance system

Hydrovision's subsidiary company Seaeye Marine offer a complete range of electric powered ROVs from the small, portable Seaeye Falcon through a series of observation, inspection and survey systems to the more capable light work class Seaeye Panther Plus.



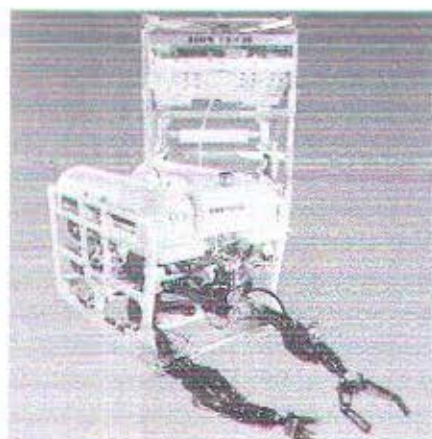
**Seaeye Falcon**



**Seaeye Lynx**



**Seaeye Surveyor Plus**

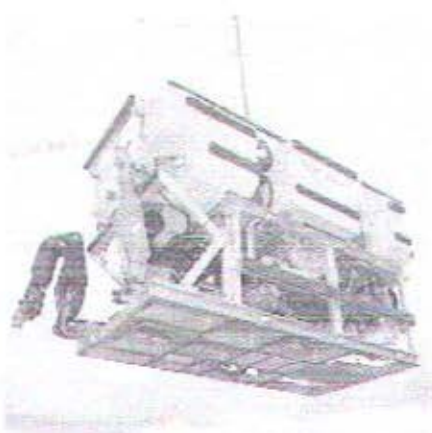


**Seaeye Panther Plus**

Provision also build customised ROVs to particular clients' requirements. These have included 75, 100 and 150 shp SCVs for Stolt Comex Seaway and a Dragon 2000 for Global Industries Inc.



**150 shp SCV - 3000 msw**



**100 shp SCV - 3000 msw**

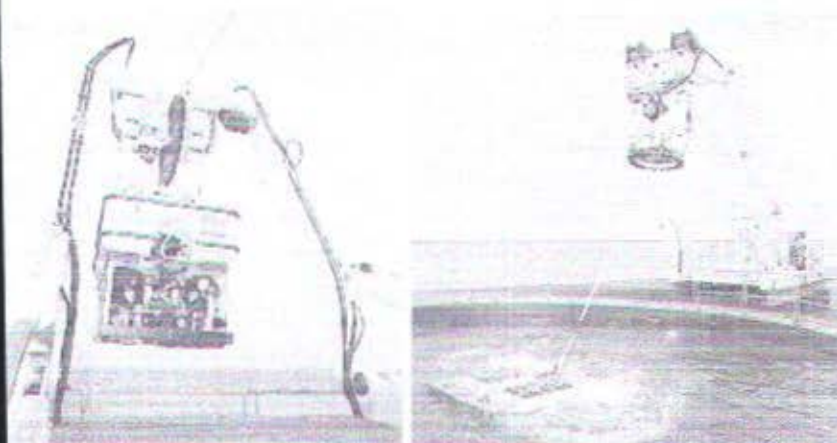
Trading of older ROVs to meet modern requirements for greater power and reliability is a provision speciality. This may include a new hydraulic system or new control system.

The company's recently expanded facilities provide an ideal location for integration and testing of systems prior to mobilisation. This service is offered to local companies who can also rent space and facilities for their own trials.



### **Launch and Recover Systems (LARS) and Handling Systems**

Hydrovision manufacture Launch and Recovery Systems for the Group's standard ROV systems as well as offering custom designed winches and handling systems for customers' special requirements.

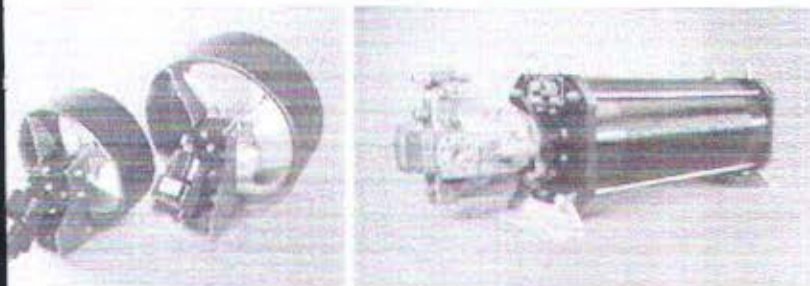


### **System Building Blocks**

Hydrovision manufacture and stock a wide range of hydraulic system and control system building blocks for customers who build their own ROVs, cable trenching systems and special subsea tools.

### **Hydraulic Building Blocks**

One of the most well known of these is the highly regarded Curvetech range of Thrusters, Power Packs, Servo Valve Packs, and Solenoid Valve Packs. New, higher efficiency MK2 Curvetech Thrusters were introduced in 1999 and a more efficient bent axis motor range of thrusters in 2000.



Other standard hydraulic products include Pressure Compensators, Reservoirs, Pan and Tilt Units, Solenoid Actuators, Check Valves and HP Filter Assemblies.



### **Control Systems**

Hydrovision's Curvetech PC104 based control system was introduced in 1999 for use in Hydrovision's next generation of work and electric powered ROVs and as a building block for other control applications. These include the control of subsea tooling, process control and as an upgrade kit for older ROVs whose control electronics have become obsolete or prohibitively expensive to support.

## Other Products

Other Hydrovision products include Subsea Flood Lights rated to 3000 metres, Video Overlays, Video Matrix Switchers and a Mini Manipulator and Cable Cutter suitable for smaller open frame electronic hydraulic vehicles.



**Demon** is a full specification work class ROV, fully interfaced for construction, intervention or survey operations. Powered by a 100 shaft hp Curveteck Propulsion System, this ROV can be operated 'free swimming' to greater depths than had previously been thought possible. Live boat operations to depths greater than 2000 metres are routine for this immensely powerful system. Alternatively Demon can be deployed from a Hydrovision long excursion tether management system, (TMS). These are available in either a 'Top Hat' or 'Garage' configurations.

The electronics, PC control system and diagnostics are a reassuring evolution of familiar technology for your pilots and technicians. This makes training easy and greatly contributes to the excellent reliability and minimal down time experienced by all Demon owners and operators. Built and supported by engineers who have worked in your business, Demon is a powerful, rugged system and built out with maintenance and easy access to components as the highest priority.

## APPLICATIONS:

### The Demon ROV



The Demon ROV can be provided in various depth ratings to 3000 metres governed by the buoyancy material fitted and rating of the electronics pods. All other components are oil filled and pressure compensated for depth. Many of our customers order their Demons fitted with 3000 metre rated electronics pods to cater for future possible requirements but have buoyancy fitted that is rated for immediate requirements. This is a sensible economy and good planning for the future. A standard Demon vehicle is provided with:

- 3 tonne through lift frame with 6 variable lift points.



- 100 shaft horse power Curvetech Propulsion System comprising:
  - Three vertical Curvetech HT 300 thrusters.
  - Four Vectored Curvetech HT 380 thrusters.
  - 100 shaft hp Curvetech double ended Power Pack rated for continuous running at full power. A single pump is fitted as standard.
  - An optional second pump can be fitted for auxiliary hydraulics.
  - 8 way Curvetech Thruster Control Unit, (TCU). Seven high flow, 77 LPM servo valves are provided for thruster control. An additional servo valve can be added for an optional, variable flow, uni-directional tooling port with a manually aligned PVR in the high pressure out let to provide variable pressure to match the tool fitted.

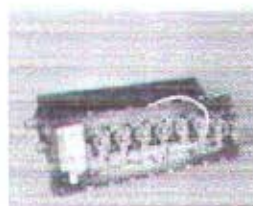
### Curvetech Propulsion System



Thruster



Power Pack



Servo Valve Pack (TCU)

- Two 10 function Curvetech solenoid valve packs.
- One heavy duty Hydrovision Pan and Tilt Unit.
- Optional look up rotary actuator.
- 1 variable pressure and flow tooling port supplied from the TCU and fitted with QDs.
- 5 power and control outlets for colour or SIT cameras.
- 6 HV 3000 under water flood lights.
- Fluxgate compass and integral rate gyro.
- Hydraulic pressure and depth sensor.
- Interface for sector scan sonar.
- Electrical penetrators for additional sensors including:
  - 3 off 24 VDC 2.5 amp switched power sources for additional sensors/tools.
  - 4 off 110 VDC 2 amp switched power sources for additional sensors / tools.
- 1 interface for transponder/responder.
- Vehicle functions / integral sensors include:
  - 96 analogue channel up and down capacity.
  - Auto heading.
  - Auto depth.
  - Trim control of thrusters.
  - Pitch and roll sensors.
  - Optional Prism fibre optic and data transmission system (This is required for umbilical lengths greater than 2500 metres).
  - Optional Auto Altitude.

### Non Vehicle Characteristics

Weight: up to 3 tonnes according to depth rating and accessories fitted



Length:	2950 mm with survey frame
Width:	1830 mm
Height:	approx 2000 mm depending on buoyancy depth rating
Payload:	250 kilos
Speed:	>3.5 knots in construction/ intervention configuration: > 4 knots in survey configuration
Ballard Pull:	>800 kgf in construction/intervention configuration

## Customer Defined Vehicle Equipment

Customers will usually specify the equipment that they wish to have fitted to their Demon vehicle. The following items would be a typical minimum fit of specified accessories that we could fit for you:

Cameras:	1 SIT, 1 Colour, 1 black and white look up camera
Sonar:	Obstacle avoidance sector scanning
Manipulators:	1 five-function rate arm (port side). This would often be used as a 'grabber' to hold onto a structure while the second, more dextrous manipulator is used to carry out the work task. One seven-function rate or position feedback arm (starboard side).

Other equipment could include a cable /rope cutter, jetting tool, survey suite and special task specific tools.

## Control System and Power Distribution Unit (PDU)

The Demon is provided with 4 bay 19 inch rack system which houses the control PC system, control monitor and pilot and observer's desks as well as two blank writing desks. The pilot and observer's control desk layouts can be customised to suit the client's preference. There is ample space in the racks for additional client specified equipment to be fitted. A 100 horse power PDU and transformer is also provided as part of the surface equipment together with inter connecting cables. The rack system, PDU and transformer can be provided free standing for installation in a vessel or in a customised control container. Containers can be zone rated or rated for safe area operations and can be conditioned if required. Standard equipment supplied includes:



- PC based control system with full ROV diagnostics monitoring.
- 14 inch graphics monitor.
- Fibre Optic video mux.
- Optional Prizm fibre optic data transmission system. This is required for umbilical lengths greater than 2500 metres.
- Power transformers

- Distribution/switching for ROV, winch and crane.
- Ground fault monitors and step up transformers to current AODC guidelines.
- 60 Hz rated, ROV electrics and HPU step up transformer tapped to accommodate input voltages from 380 to 480 volt steps.
- Local isolation of winch, ROV and crane. Voltage, frequency and current meters. HPU hours run meter and phase rotation indicator.

## Client Defined Surface Control Equipment

The following additional equipment will usually be specified by the client for installation into the surface control rack system:



- Video Monitors - 14 inch and, or twin 9 inch screens.
- Video Recorders - usually three S-VHS units.
- Hydrovision VO 2000 Video overlay unit.
- Hydrovision 16 x 16 video matrix switcher.
- Sonar display monitor.
- Audio communications system

Other equipment may include Bathymetric systems (UK 94), ROV tracking system, cable and pipe marker. We can install all of these for you.

## Launch, Recover and Deployment Equipment

The enormous power of the Demon's Curvetech propulsion system allows free swimming and live operations to be conducted far deeper than has previously been thought possible. A free swimming Demon at 2000 metres depth is not uncommon. Hydrovision can offer launch and recover equipment for this type of operation.

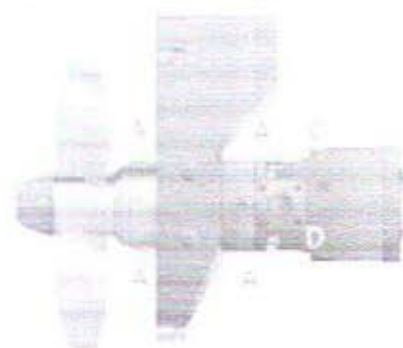
- Umbilical or lifting winches.
- Soft or armoured umbilicals.
- 'A' Frame or Crane packages including umbilical sheaves and lock latch systems.

For deeper operations or conditions that call for a reduction of the effect of water current on the ROVs, Hydrovision can supply their 'Top Hat' or 'Garage' configuration of long excursion Tether Management System (TMS).



The MK2 Curveteck Thruster is up to 11 percent more efficient than the original Curveteck HT 300s it introduced in 1993. Fitted with a axial piston motor (AP) this thruster can be supplied in various capacities by adjustment and fixing the swash angle. It is the preferred choice for upgrading older ROVs or for new build 75 and 100 shp systems where space to fit the more efficient HT 380 may be restricted. It is widely used as a vertical thruster on 100 shp ROVs. The most common capacity supplied is 23 cc/rev with a 14 degree swash angle.

#### Installation Details



#### Details:

Bearing lubrication oil return and bleed ports:	6 off 1/8" BSPP
Motor case drain:	4 off 1/4" BSPP
Motor supply and return:	3/4" BSPP 4 holes 1/4"
	BSF on 44.9 PCD

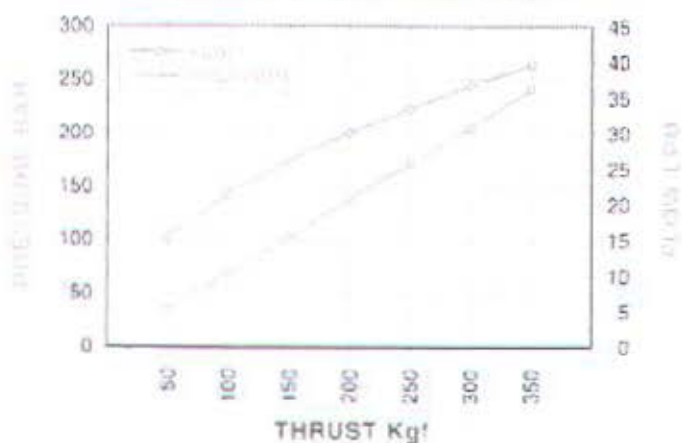


#### Specifications:

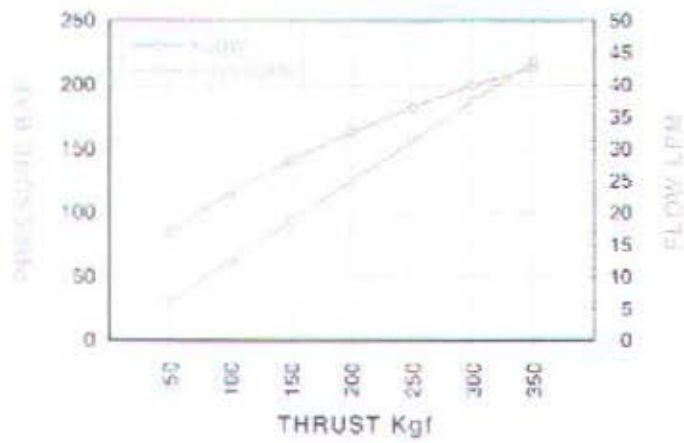
Maximum Static Thrust	>300 Kgf forward and Kgf reverse
Maximum Operating Pressure	280 bar
Hydraulic flow	See performance graph
Propeller Diameter	300 mm
Mounting Type	Pedestal
Weight in Air	21 Kg ( 46 lbs)
Weight in Water	12.5 Kg

#### Performance

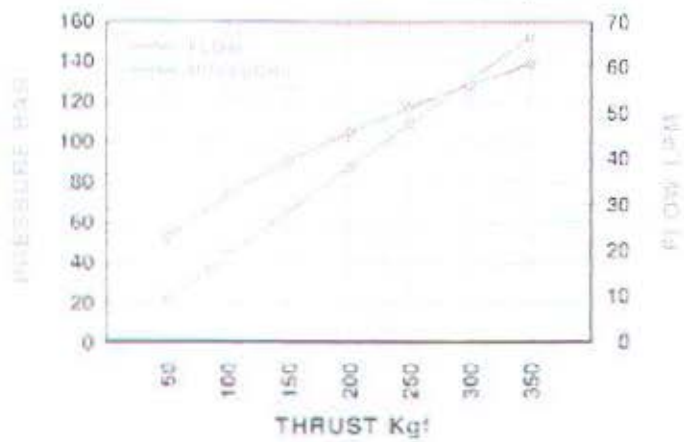
MK2 HT 300AP-13 at 0 Knots  
13° Swash Angle = 21 cc / Rev Displacement  
Pressure / Thrust / Flow Graph



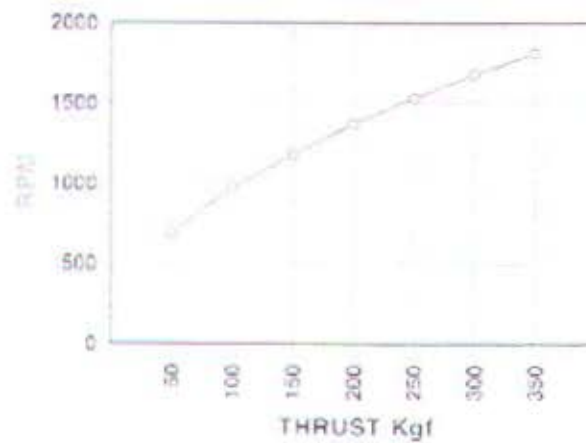
MK2 HT 300AP-14 at 0 Knots  
 14" Swash Angle = 23 cc / Rev Displacement  
 Pressure / Thrust / Flow Graph



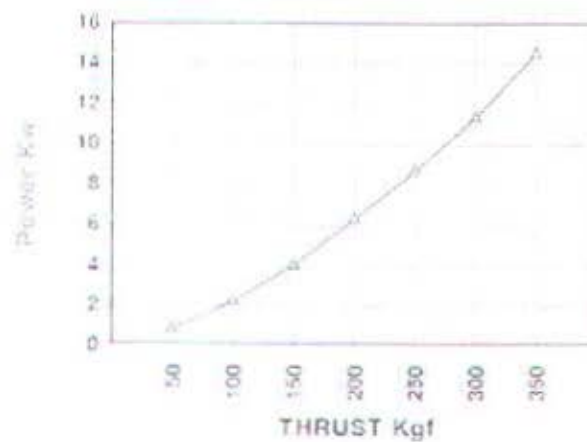
MK2 HT 300AP-20 at 0 Knots  
 20" Swash Angle = 33 cc / Rev Displacement  
 Pressure / Thrust / Flow Graph



MK2 HT 300AP-14 at 0 Knots  
 14° Swash Angle = 23 cc / Rev Displacement  
 RPM v Thrust



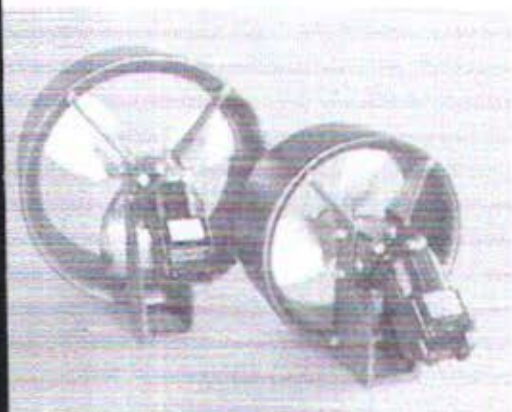
MK2 HT 300AP-14 at 0 Knots  
 14° Swash Angle = 23 cc / Rev Displacement  
 Power Curve



2 HT 300AP

The development of the popular HT 300 thruster size uses a lightweight, fixed displacement, cc/revolution aluminium motor specifically developed for Hydrovision that is interfaced to the standard HT 300 bearing housing.





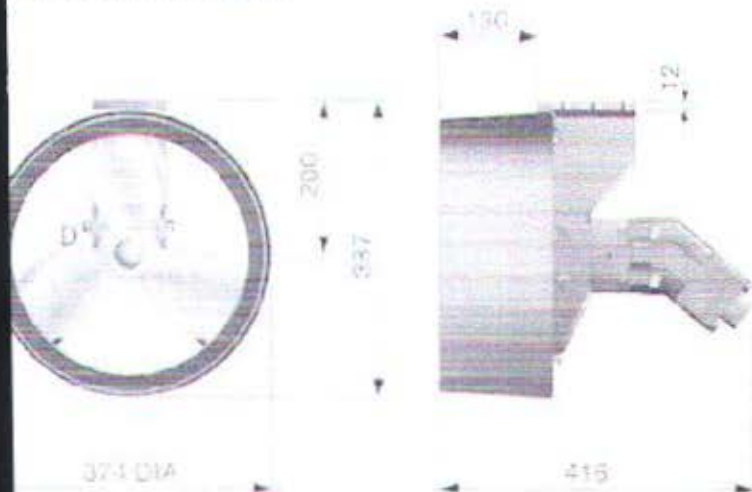
Survetech HT 380BA and HT 300BA Thrusters

is motor offers the benefits of:

- Improved efficiency
- Lighter weight
- Lower cost

## Survetech HT 300BA-23

### Installation Details



Pedestal Mounting Detail

(All dimensions in mm)



View with nozzle removed

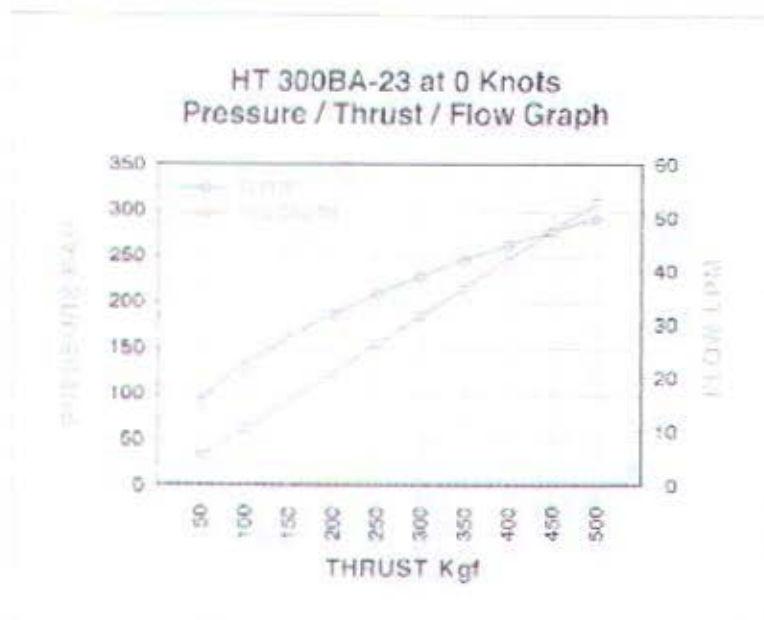
#### Port Details:

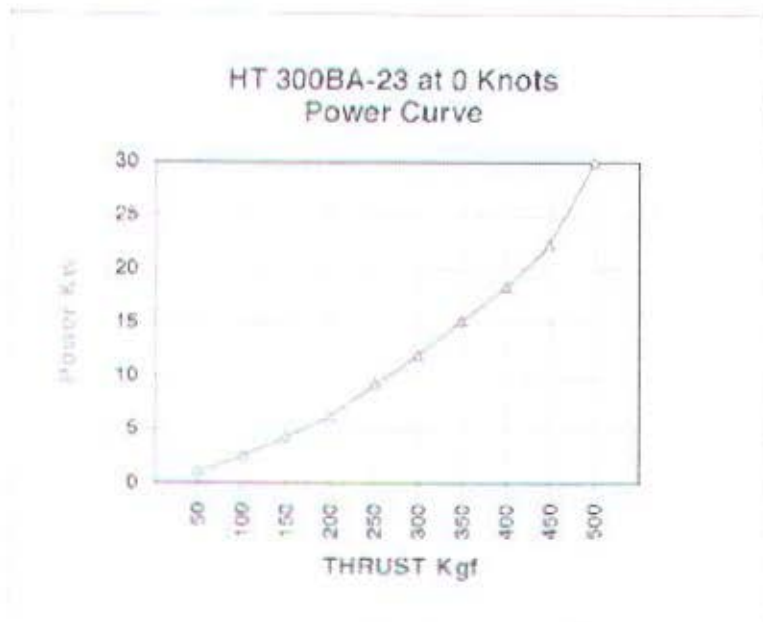
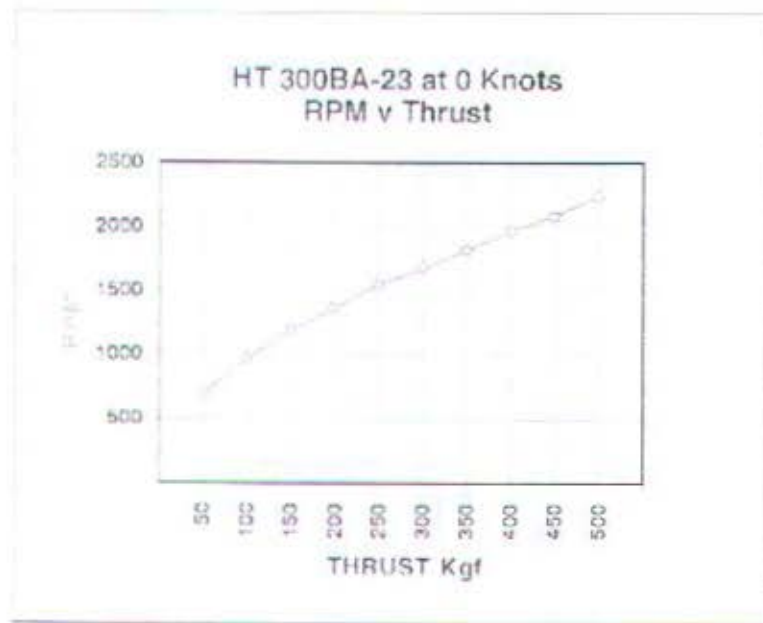
Bearing lubrication oil return and bleed ports:	6 off 1/8" BSPP
Motor case drain:	2 off 3/8" BSPP
Motor supply and return:	3/4" BSPP
Bleed Ports	3 off 1/8" BSPP

#### Specifications:

Maximum Static Thrust	300 Kgf at 186 bar and 39 LPM (forward and reverse)
Maximum Operating Pressure	300 bar
Hydraulic flow	See performance graph
Propeller Diameter	300 mm
Mounting Type	Pedestal
Weight in Air	15.5 Kg
Weight in Water	7 Kg
Motor Capacity	23cc/revolution

#### Performance





## 300BA-23

This thruster shares the same shaft, bearing housing and motor as the Curvetech HT 300AP but has a larger propeller, nozzle and pedestal. It develops higher torque and greater thrust for the same hydraulic flow as the HT 300AP and so is significantly more efficient. Most modern 100 and 150 shp boats will be fitted with this thruster in a vectored configuration to achieve bollard pulls in excess of 500 kgf.

## Installation Details





Figure 1: Motor Dimensions

Table 1: Motor Dimensions

Table 2: Motor Dimensions

# **Port Details:**

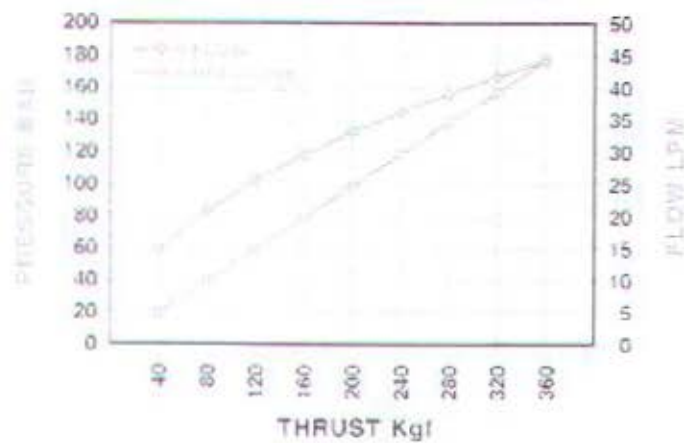
Bearing lubrication oil return and bleed ports:	6 off 1/8" BSPP
Motor case drain:	4 off 1/4" BSPP
Motor supply and return:	3/4" BSPP 4 holes 1/4" BSF on 44.9 PCD

# **Specifications:**

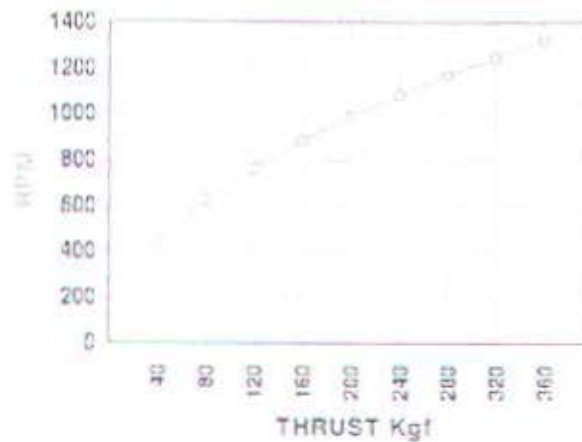
Maximum Static Thrust	>360 Kgf forward and Kgf reverse
Maximum Operating Pressure	280 bar
Hydraulic flow	See performance graph
Shaft Diameter	380 mm
Mounting Type	Pedestal
Weight in Air	30 Kg ( 66 lbs)
Weight in Water	15 Kg

# **Performance**

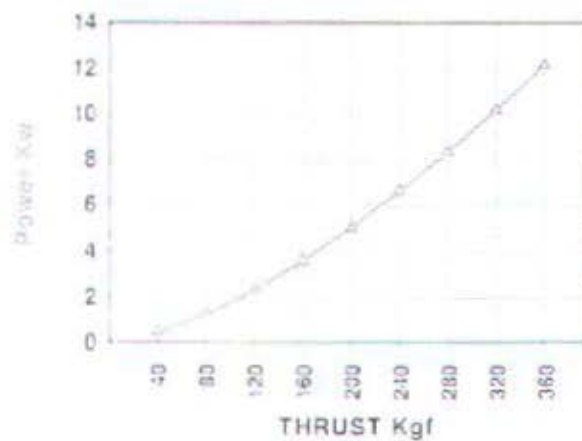
MK2 HT 380AP-20 at 0 Knots  
 20° Swash Angle = 33cc / Rev Displacement  
 Pressure / Thrust / Flow Graph



MK2 HT 380AP-20 at 0 Knots  
 20° Swash Angle = 33 cc / Rev Displacement  
 RPM v Thrust

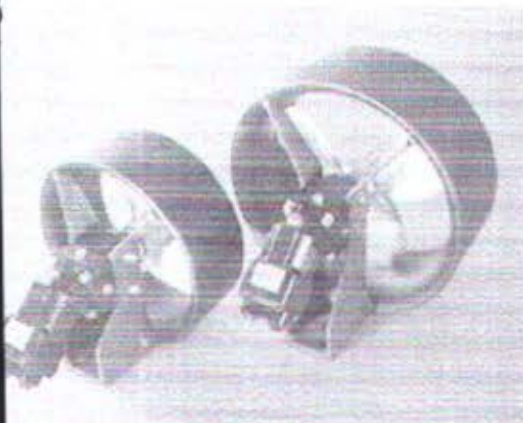


**MK2 HT 380AP-20 at 0 Knots**  
**20° Swash Angle = 33 cc / Rev Displacement**  
**Power Curve**



## MK2 HT 380AP

The Curvetech HT 380BA-32 is a development of the best selling HT 380AP thruster. It uses a lightweight, fixed displacement, 32cc/revolution, Bent Axis aluminium motor specially developed for hydrovision. This motor interfaces with the standard HT 380/300 bearing housing and can be used to upgrade HT 380AP (Axial Piston) motors with some savings in cost.



**Curvetech HT 300BA and HT 380BA Thrusters**

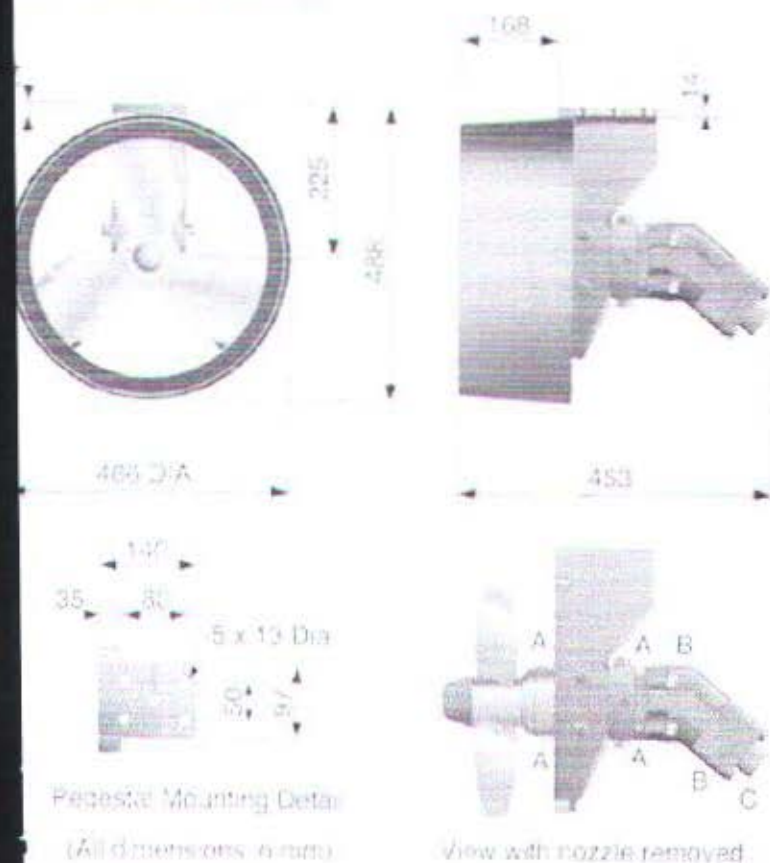
The motor provides the Curvetech HT 380 BA-32 with the benefits of:

- Greater efficiency
- Lighter weight
- Lower cost



## Curveteck HT 380BA-32

### Installation Details



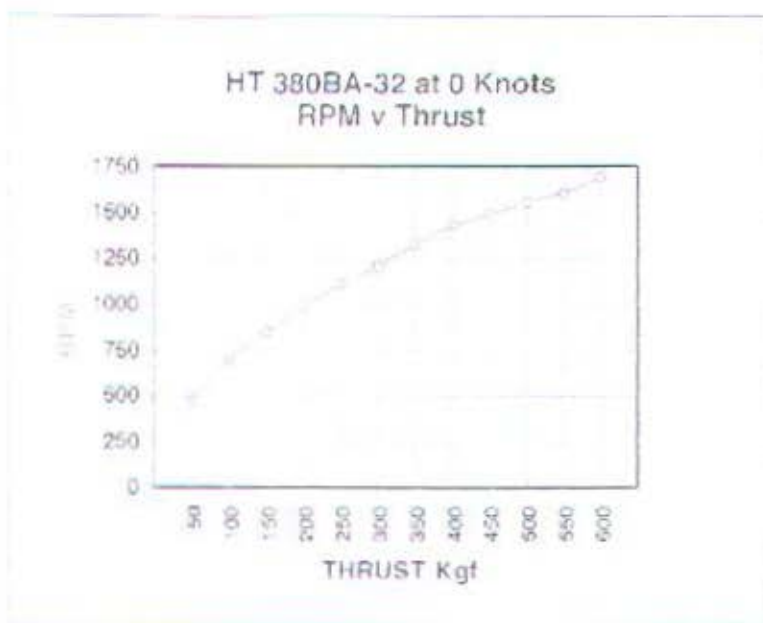
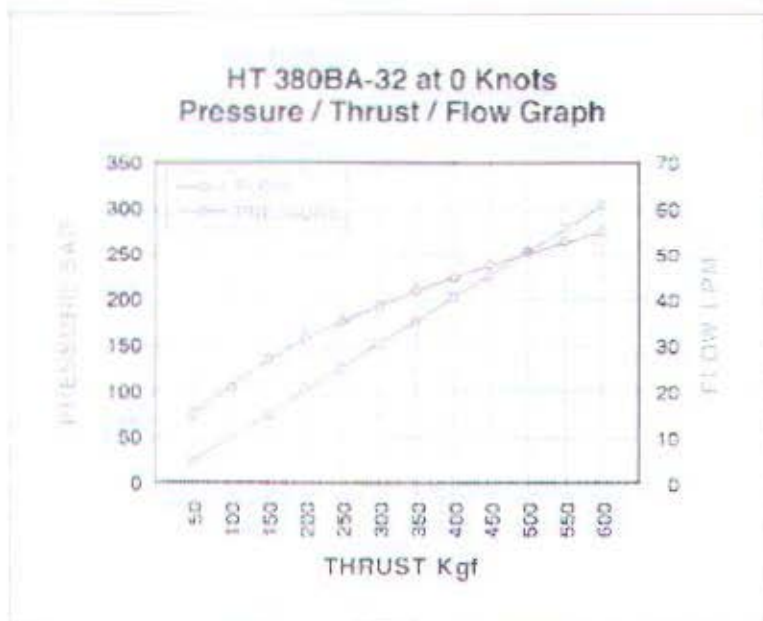
### Port Details:

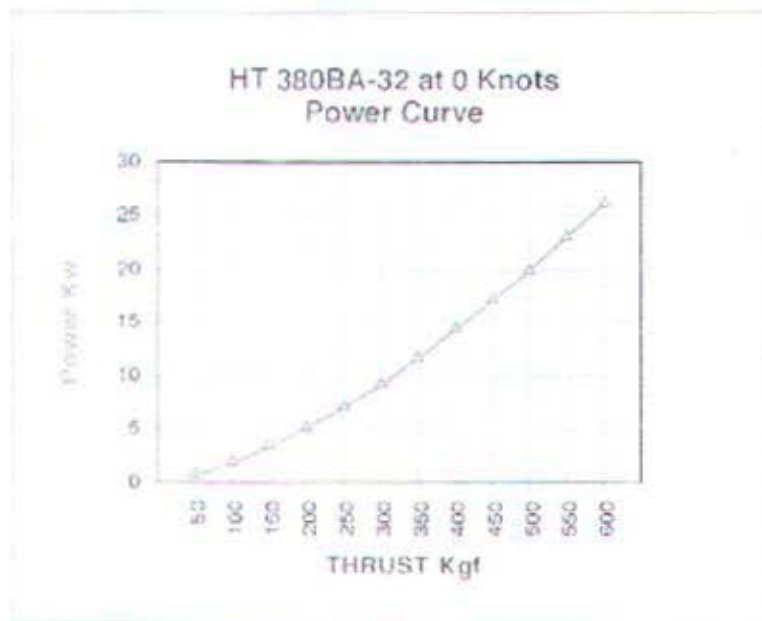
Bearing lubrication oil return and bleed ports:	6 off 1/8" BSPP
Motor case drain:	2 off 3/8" BSPP
Supply and return:	2 off 3/4" BSPP
Bleed Ports	3 off 1/8" BSPP

### Specifications:

Maximum Static Thrust	390 Kg at 186 bar and 41LPM (forward and reverse)
Maximum Operating Pressure	300 bar
Hydraulic flow	See performance graph
Cylinder Diameter	380 mm
Mounting Type	Pedestal
Weight in Air	24.5 Kg
Weight in Water	9.5Kg
Displacement Capacity	32cc/revolution

performance





### 380BA-32

**CURVETECH POWER PACKS** are designed for subsea use and are oil filled and separately pressure compensated.

Applications include:

- ROV hydraulic power packs
- TMS hydraulic power packs
- Tooling hydraulic power packs
- LP and HP Water Jetting power packs for pipe and cable burial tools

Power pack failures are a major cause of system down caused by the failure of the electric motors. Curvetech power packs solve this problem and have proved so reliable that we now offer a two-year warranty on the electric motors that we use.

No compromise in reliability or performance is acceptable for a Curvetech product and our power packs are no exception. Every unit manufactured is fully tested and certified for continuous running at full power.

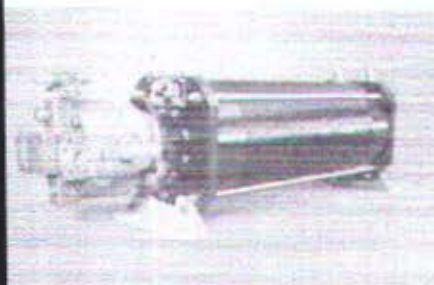
A wide range of standard sizes is available and each share a unique motor winding technique to ensure exceptional cool running. This technique was first developed for a particularly compact power pack required by the US Navy.

Temperature sensors installed within the motor windings monitor the power pack temperature. This gives you much better warning of any overheating of the motor than the common alternative practice of monitoring the cooling oil temperature.

The winding insulation is specified to accept either hydraulic or transformer oil without damage and both, concentric running and long life is ensured by oversize bearings.



Most Curvetech power packs are dual ended to accommodate the option of two hydraulic pumps. For ROV applications this allows the useful option of separating propulsion and tooling hydraulics into two separate circuits. Industry standard pumps can be fitted using an appropriate adaptor kit incorporating flexible coupling and flange.



10 shp motor fitted with Rexroth RVS0140 Pump

- Two year motor warranty
- Rated and tested for continuous running at full power
- Cooler running for longer life
- Oversize bearing for reliability and smooth running
- Accepts transformer or hydraulic oil for convenience
- Accepts choice of standard pumps for versatility
- Water ingress and motor winding temperature sensors for reassurance
- Double ended motor for optional second pump
- Wide range to meet most applications
- Special designs accommodated
- Another Curvetech product for complete confidence

#### Applications

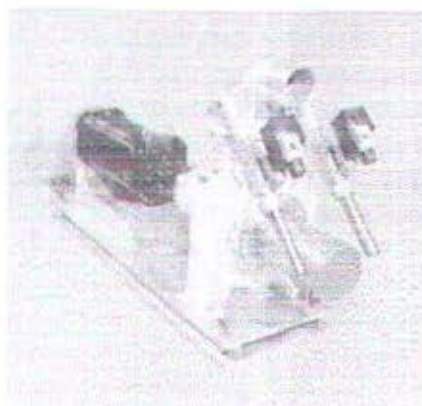
Curvetech Power Packs are widely used on the world's most power ROVs including Demon, Diablo, Glass, Sabre, SCVs, Explorer and Dragon, and are the heart of their Curvetech Propulsion systems.

For ROV and most tooling applications, we recommend the reliability of 1800 rpm, 4 pole power packs and 3000 or 2400 volts. 1100 volt power packs can be provided but are less common these days as ROV operations move into deeper water and operate at higher voltages.

For HP water jetting applications, 2 pole motors are more widely used coupled to High Pressure water pumps which operate at 3600 rpm. 4 pole motors can be used for the LP water pumps.



125 shp Curvetech motor



4 pole 75 shp Curvetech motor

and Gargarino LP MU 200-250  
water pump

order a complete Curveteck Hydraulic Power Pack it is necessary to select:

- The required shaft horse power of the oil filled electric motor
- A hydraulic pump matched to the motor size
- An auxiliary pump if this is required
- The appropriate pump to motor interface kits and
- A bulkhead power connector or hard wiring in a compensated oil tube

A separate pressure compensator will also be required to maintain the pressure of oil in the motor at 1 (15 psi) above ambient to depth. Hydrovision's EV 4001, 1.42 litre (2.5 pint) compensator would be appropriate and can be supplied with an analogue or reed switch alarm. (See Pressure compensators).



Copyright © 2004 John Wiley & Sons, Ltd.

POLE (MBC)		WT. IN AIR	WT. IN WATER	F	G	H	K	L	M
4	532	110kg	83kg	283	302	350	274	130	180
4	670	180kg	130kg	310	315	365	300	130	210
4	799	200kg	145kg	310	346	365	300	130	310
4	830	240kg	180kg	310	315	365	300	130	210
2	720	205kg	160kg	310	315	365	300	130	210
2	810	245kg	185kg	310	315	365	300	130	210
4	914	382kg	285kg	383	398	460	374	130	280







FIGURE 10-10 HYDRAULIC PUMP

PUMP		TYPICAL PUMP OPTIONS					
Interface	Manufacture	Type	cc/rev	B	C	D	E
Part						(Pressure	(Suction
No.						Port)	Port)
T 4017-2	REXROTH	A10VSO-45	45	197	237	1" SAE 3000#	1 1/2" SAE 3000#
T 4017-4	REXROTH	A10VSO-100	100	299	350	1 1/4" SAE 6000#	2 1/2" SAE 3000#
T 4017-5	ROTARY POWER	A10VSO-140	140	321	383	1 1/4" SAE 6000#	2 1/2" SAE 3000#
T 4017-6	ROTARY POWER	A70-LG	11.5	1	129	1/2" BSPP	RHL SPECIAL
T 4017-7	ROTARY POWER	A200-LG	33	13	183	3/4" BSPP	RHL SPECIAL
T 4017-8	ROTARY POWER	A380-LG	62	56	270	1" SAE 3000#	1" SAE 3000#
T 4017-9	ROTARY POWER	A560-LG	92	56	270	1 1/4" BSPP	RHL SPECIAL
4017-10	ROTARY POWER	A760-LG	125	86	360	1 1/2" SAE 6000#	1 1/2" SAE
4017-14	REXROTH	A4VSO-180	180	377	438	1 1/4" SAE 6000#	3" SAE 3000#
4017-16	REXROTH	A10VSO-28	28	166	208	3/4" SAE 3000#	1 1/4" SAE 3000#

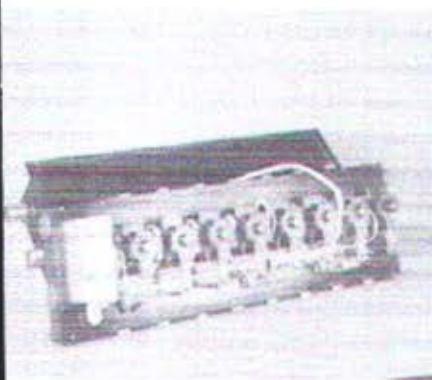
**Curvtech Servo Valve Packs** are available in a number of standard configurations and have mainly been supplied for the variable speed control of hydraulic thrusters fitted to ROVs. In this application they are often referred to as a 'TCU' or Thruster Control Unit. Regular and Compact sizes are offered as well as custom designed units.

Servo Valve Packs can also be used to control any uni or bi-directional hydraulic function or actuation also requires remote control of that function's speed. Remote control of a hydraulic rotary cutter, grinder or water pump would be typical applications.

Selection of the appropriate rating of the servo valve to be fitted is an important consideration in order to avoid an excessive pressure drop across the valve when full flow is demanded.

## 008 CURVETECH 8 FUNCTION SERVO VALVE PACK





Used as the standard TCU on Demon, Diablo, Dragon 2000 and Explorer ROVs, this valve pack forms part of their Curvetechn Propulsion System. It is ideally matched to control Curvetechn HT 300, HT 380 and HT 420 thrusters in conjunction with a Curvetechn 100 or 150 shaft horsepower power pack.

[Click here to view CAD drawing of DI 6008 - Curvetechn 8 Function Servo Valve Pack](#)

The DI 6008 Curvetechn servo valve pack manifold can accommodate and be ordered with up to 8 valves fitted. When used as a TCU one valve will be fitted for each thruster and additional valves used to remotely control a variable flow tooling outlet. See an example circuit under ROV Hydraulic Systems on this web site.

Log flapper type servo valves are used which are rated to 77 LPM and 68 Bar pressure drop at full flow. When used to control an HT 300 or HT 380 Curvetechn thruster, whose maximum demand flow is 100 LPM at 200 Bar, the maximum pressure drop across the valve will be a low 19.6 Bar.

Mounted at the manifold is a combined soft start and over pressure relief valve block. The soft start valve is designed to off-load the electric motor on initial start up and is completely automatic, with no electrical input. The over pressure relief valve provides system protection in the event of failure of the pump pressure compensator and is fully adjustable.

The manifold low-pressure gallery is fitted with a temperature sensor providing constant temperature monitoring of the return oil.

The valve pack lid cavity is isolated from the main system and separately pressure compensated. This allows the lid to be drained and removed on deck for maintenance of the valve pack without disturbing the main system hydraulics.

The standard TCU is fitted with a 12-pin Burton connector.

#### **Hydraulic ports include:**

Pressure Gallery - 2 X 1 1/4" BSPP

Return Gallery - 2 X 1 1/4" BSPP

Supply Ports - 3/4" BSPP

Bleed Ports - 3/8" BSPP

Filled Weight in Water 35 Kg

Filled weight in air 75 Kg

#### Ordering Information:

6008-1	Curvetech 6 Function Servo Valve Pack (with two valve stations blanked)
6008-2	Curvetech 7 Function Servo Valve Pack (with one valve station blanked)
6008-3	Curvetech 8 Function Servo Valve Pack (all valve stations are used)

#### P 6001 CURVETECH 8 FUNCTION SERVO VALVE PACK (with Remote Pump Control)

Hydrovision designed this Curvetech Servo Valve Pack for use on Stolt Offshore's SCV 3000-100 and V 3000-150 shaft horsepower ROVs as a Thruster Control Unit (TCU). It will also be used in the new generation of Hydrovision ROVs.

The specification of this unit is the same as the DI 6008 series except the system soft start valve mounted on the manifold is replaced with a pump control poppet valve. This allows the power pack hydraulic pump pressure to be reduced to a nominal 27.5 Bar, 400 psi, on system start up. This can also be used to remotely switch the pump to 'idle' when the vehicle is parked on the seabed between dives, in a TMS or on deck and will extend the life of the power pack hydraulic pump.

[Click to view CAD Drawing of VP 6001 - Curvetech 8 Function Servo Valve Pack \(with Pump Control\)](#)

#### CURVETECH COMPACT SERVO VALVE PACKS

Curvetech Compact Servo Valve Packs are available in 6 and 7 function versions as:

6030	Curvetech Compact Servo Valve Pack - 6 Function
6010	Curvetech Compact Servo Valve Pack - 7 Function



010



These units are ideal where space and weight are at a premium and a priority over a marginal loss in performance due to a slightly increased pressure drop across the valves when operating at full flow. They have been provided for use as a TCU to control Curvtech Thrusters on 50 and 75 shaft horsepower ROVs build by Mitsubishi Heavy Industries for the Japanese Defence Forces.

The size and weight reduction is achieved by using narrow body Star flapper type servo valves that can be specified to a maximum flow of 60 LPM.

A self-contained block mounted on the manifold holds the soft start valve and the system over pressure relief valve. The soft start valve is designed to off-load the electric motor on initial start up and is completely automatic, with no electrical input.

The over pressure relief valve provides system protection in the event of failure of the pump pressure compensator and is fully adjustable.

A temperature well fitted in the manifold low-pressure gallery is the temperature well providing constant temperature monitoring of return oil. The valve pack lid cavity is isolated from the main system and is separately pressure compensated. This allows valve maintenance and adjustments to be carried out without disturbing the main system.

A standard unit is fitted with a 12-pin Burton connector.

**Hydraulic ports include:**

Gallery - 2 X 1 1/4" BSPP

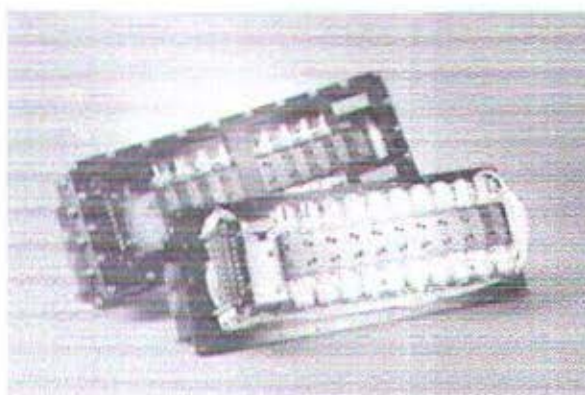
Gallery - 2 X 1 1/4" BSPP

Supply Ports - 3/4" BSPP

Bleed Ports - 3/8" BSPP

[Click to view CAD Drawing of VP 6030 - Curvtech Compact Servo Valve Pack - Function](#)

[Click to view CAD drawing of VP 6010 Curvtech Compact Servo Valve Pack - Function](#)





Curveteck solenoid valve packs in 4, 6, 8, or 10 function versions are intended for subsea use and were originally developed to operate the auxiliary hydraulic functions on work class ROVs. This includes rate controlled manipulators, pan and tilt units and rotary and linear actuators. Customised valve packs can also be provided.

Standard valve packs are provided with:

- **An Integral Pressure Reducing Valve**  
The PRV sets the supply pressure to the solenoid valves. This eliminates the need to fit an in-line PRV exposed to seawater.
- **Solenoid Valves**  
Direct operated open centre spool valves with 4 ports and 3 operating positions. When the spool is de-energised it is spring returned to the centre off position. The valve is mounted on top of the dual pilot check and the relief valve and the spools are vented to allow subsea operation.
- **Dual Pilot Check and Relief Valves**  
The dual pilot check and relief valve acts as an interface between the solenoid valve and the hydraulic equipment. This provides negligible leakage and protection for the equipment, hoses and valve from over pressurisation due to external forces.
- **In-Line Flow Controllers with Integral Check Valves**  
The flow controllers allow individual setting of the speed of movement in both directions for each controlled hydraulic function.
- **Manifold Cover Over Pressure Relief Valve**  
A relief valve set at 1.4 bar (20psi) protects the cover in the event of a high pressure leak.
- **Diode Steering**  
Diode steering allows control of a valve from a single +/- 24V DC control line. This results in a reduced number of electrical connections to the valve pack compared with the conventional valve directional control which requires 2 control lines per valve.
- **Diagnostics LEDs**  
The LEDs are visible through the clear polycarbonate valve manifold cover.
- **Water Ingress Alarm**

#### SPECIFICATIONS Curveteck Solenoid Valve Packs:

##### 6050 10 Function Curveteck Solenoid Valve Pack, 207 bar (3000 psi) (3800psi cracking pressure)

Complete with:

OV 4423	12 way bulkhead connector (Burton)
OV 4526	Pressure reducing valve, 69 to 207 bar (1000 - 3000 psi)
VP 8012	Diode steering / LED PCB assembly
VR 9043-1	Solenoid valve 4-way 3-position, 24V DC (10 off)
VR 9042-3	Dual pilot check and crossover relief valve - 207 bar (3000 psi) (3800 psi cracking pressure) (10 off)
OF 8009	Manifold cover over pressure relief valve, 1.4 bar (20 psi)
OF 8017	Bi directional flow controllers (20 off)

##### 051 10 Function Curveteck Solenoid Valve Pack, 69 bar (1000 psi) (1450psi cracking pressure)

Complete with all of the above but with

VR 9042-1 Dual pilot check and crossover relief valve  
69 bar (1000 psi), (1450 psi cracking  
pressure) (10 off)

Weight in Air 21 Kilos

Weight in Water 8 Kilos

Dimensions(mm) see sketch

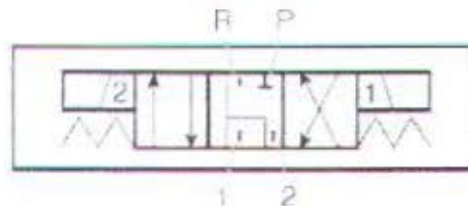
Materials 316 SS fittings, 6082-T6 hard anodised black aluminium  
manifold cover with polycarbonate observation window.  
Ports for pipe connections fitted with plastic transport  
plugs.

Installation Manifold cover to be separately compensated at up to 1  
bar (15 psi) or connected to system reservoir.

#### Standard Solenoid Valve

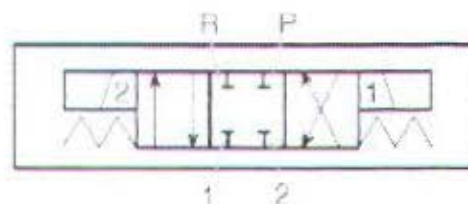
9043-1 Type Open centre  
Operating Pressure 69 to 207 bar (1000 - 3000 psi)  
Operating temperatures 2°C to 50°C (35°F - 120°F)  
Rated flow 15 lpm (4 US GPM) at 28bar (400psi) P  
Voltage 24±4 VDC  
Power 20 Watts at 28VDC

Schematic



#### Alternative Solenoid Valves

9043-2 Type Closed centre  
(used without pilot check and crossover  
relief)  
Other Specifications As VR 9043-1  
Schematic

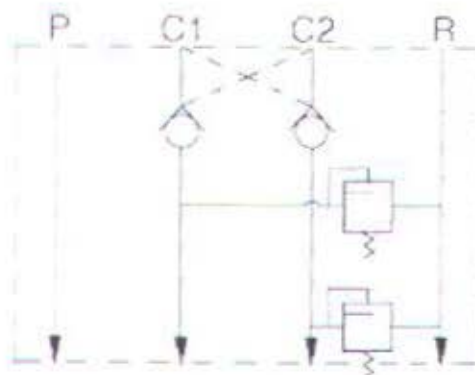


9043-4 Type Open centre  
Voltage, Power 110±10VDC, 20 watts@ 120 VDC

P 9043-3	Type	Open centre
	Voltage, Power	12± 2 VDC, 20 watts@ 14 VDC
P 9043-3	Type	Open centre
	Voltage, Power	24± 4 VDC, 15 watts@ 18 VDC

#### Standard Dual Pilot Check and Crossover Relief Valves

9042-3	Used on	VP 6050 and any other 207 bar (3000psi) solenoid valve packs
	Operating pressure	207 bar (3000 psi)
	Cracking pressure	262 bar (3800 psi)
9042-1	Used on	VP 6051 and any other 68 bar (1000 psi) solenoid valve packs
	Operating pressure	69 bar (1000 psi)
	Cracking pressure	100 bar (1450 psi)
9042-2	Used on	Optional for 138 bar (2000 psi) solenoid valve packs
	Operating pressure	138 bar (2000 psi)
	Cracking pressure	165 bar (2400 psi)
	Schematic	



6050/VP 6051 10 Function Solenoid Valve Pack 207/69bar





Figure 1: 4/3-way solenoid valve pack



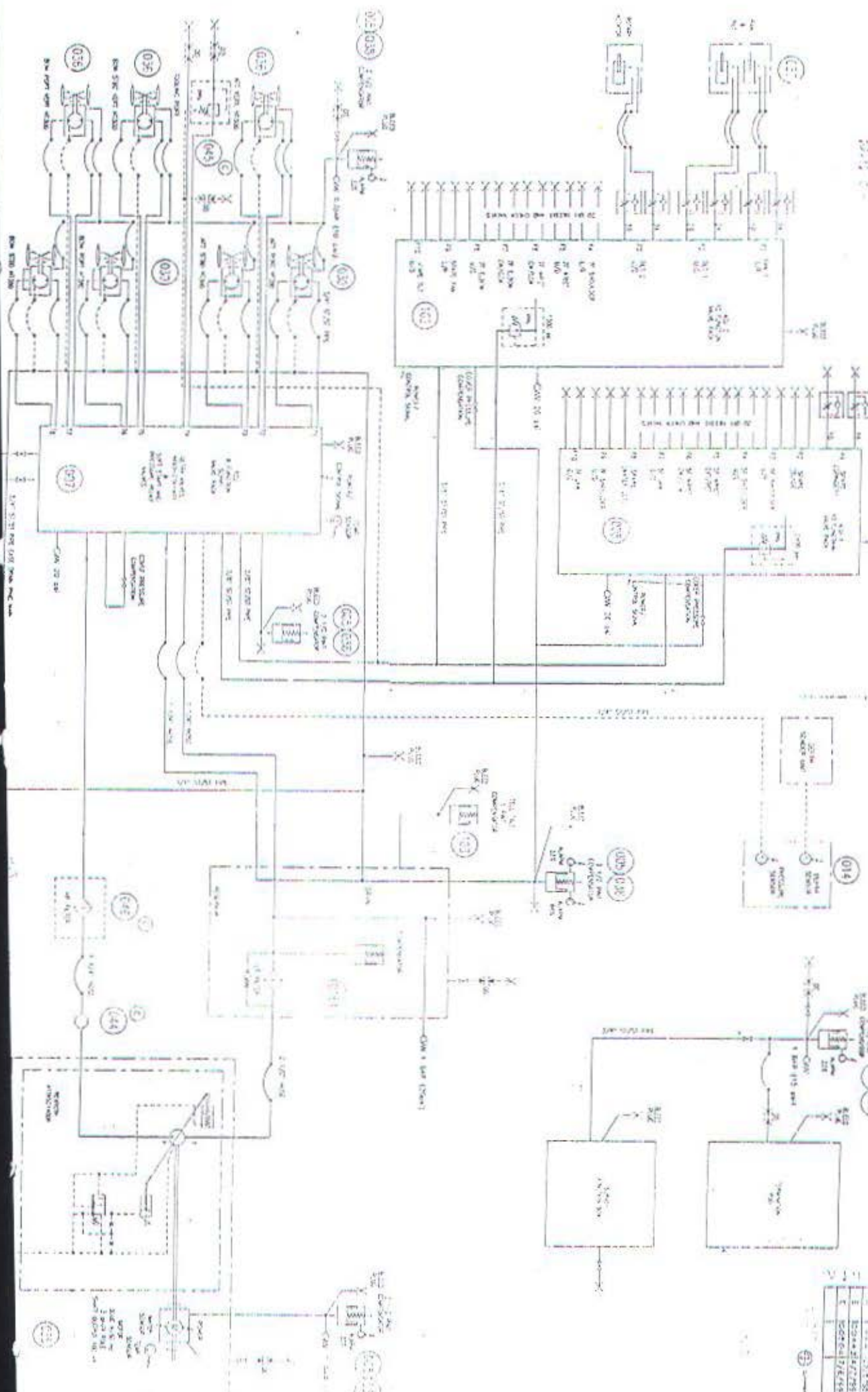
Figure 2: 4/3-way solenoid valve pack

## ATIONS

vetech Solenoid Valve Packs can be provided in alternative configurations to suit your particular requirements. For example:

- Hard wired without diode steering
- Without pilot checks and cross over reliefs
- With closed centre valves
- With 68, 138 or 207 bar (1000, 2000 or 3000 psi) maximum operating pressures determined by selection of the pilot and cross over relief rating.
- Alternative valve operating voltages (12 or 110 VDC)
- 4,6 or 8 Function Solenoid Valve Packs





Item	Qty	Part No.	Description
1	1	012	Relay
2	1	013	Relay
3	1	014	Relay
4	1	015	Relay
5	1	016	Relay
6	1	017	Relay
7	1	018	Relay
8	1	019	Relay
9	1	020	Relay
10	1	021	Relay
11	1	022	Relay
12	1	023	Relay
13	1	024	Relay
14	1	025	Relay
15	1	026	Relay
16	1	027	Relay
17	1	028	Relay
18	1	029	Relay
19	1	030	Relay
20	1	031	Relay
21	1	032	Relay
22	1	033	Relay
23	1	034	Relay
24	1	035	Relay
25	1	036	Relay
26	1	037	Relay
27	1	038	Relay
28	1	039	Relay
29	1	040	Relay
30	1	041	Relay
31	1	042	Relay
32	1	043	Relay
33	1	044	Relay
34	1	045	Relay
35	1	046	Relay
36	1	047	Relay
37	1	048	Relay
38	1	049	Relay
39	1	050	Relay
40	1	051	Relay
41	1	052	Relay
42	1	053	Relay
43	1	054	Relay
44	1	055	Relay
45	1	056	Relay
46	1	057	Relay
47	1	058	Relay
48	1	059	Relay
49	1	060	Relay
50	1	061	Relay
51	1	062	Relay
52	1	063	Relay
53	1	064	Relay
54	1	065	Relay
55	1	066	Relay
56	1	067	Relay
57	1	068	Relay
58	1	069	Relay
59	1	070	Relay
60	1	071	Relay
61	1	072	Relay
62	1	073	Relay
63	1	074	Relay